mikro számítógép magazin

Ára: 30 Ft

III. ORSZÁGOS MIKROSZÁMÍTÓGÉPES TALÁLKOZÓ

1988/3

μ '88
III. ORSZÁGOS MIKROSZÁMÍTÓGÉPES TALÁLKOZÓ
március 19—26.

A találkozó A SZÁMÍTÁSTECHNIKA MINDENKIÉ, A SZÁMÍTÁSTECHNIKA MINDENKIÉRT informatikai kiállítás köré épül. Hagyományosan jelen lesznek az NJSZT területi szervei, a Hobby Computer Club és a számítástechnika számos híres egyénisége.

Érdekes versenyek, vetélkedők, rendezvények!

Az oktatóprogramok egyéni és országos csapatversenye a legkülönbözőbb témákban, nemcsak pedagógusok számára indul. Látványos és műszakilag is érdekes a házi építésű számítástechnikai eszközök országos versenye. Rendszerek, perifériák és számítógéppel vezérelhető modellek, játék vasutak, autók és robotok mérik össze tudásukat.

A „Sütés-főzés számítógéppel" vetélkedőn azok indulnak, akik valamilyen módon felhasználják a számítógépet ételek, italok készítéséhez.

Újdonság a találkozón „Az ember elgépiesedik — a gép elemberiesedik" című karikatúrapályázat, amely hagyományos eszközökkel és számítógéppel készített karikatúrák kiállítása. A rendezvényen a látogatók is készíthetnek gunyoros rajzokat, és nemzetközi zsűri által értékelt karikatúrákból összeállított, reprezentatív albumot nyerhetnek vagy vásárolhatnak.

A találkozó egészségügyi szekciója a számítástechnikával foglalkozó mozgáskorlátozottak, vakok és gyengénlátók táborát reprezentálja. Bemutatják az első, vakok számára kifejlesztett számítástechnikai munkahelyet, és mindenki kipróbálhatja a KFKI-ban készült „beszélő" számítógépet. Itt ismerkedhetnek meg a látogatók „A rák ellen — az emberért, a holnapért" alapítvánnyal is.

Az olvasó—szerkesztő találkozón az érdeklődők elbeszélgethetnek a számítástechnikai lapok szerkesztőivel.

A rendezők azokra is gondoltak, akiknek feltalálói hajlamaik vannak. A „Szereld össze" akció során egy zacskóban kapott alkatrészhalmazból kell valamilyen „működő" berendezést összállítaniuk az erre vállalkozóknak.

A sakkozók a legújabb gépekkel mérhetik össze tudásukat. Gép géppel is versenyez: kié a legjobb sakkozóprogram?

Hazánkban első ízben itt tekinthetik meg az érdeklődők a számítástechnikai eszközök múzeumát.

A kiállításon a számítástechnikában élenjáró cégek mutatják be újdonságaikat, melyekből vásárolni is lehet.

A rendezvényen kedvezményesen árusítják a Mikroszámítógép Magazin régebbi számait, és előfizetést is elfogadnak.

A μ '88-at nem szabad kihagyni!

μ mikro számítógép magazin

6. ÉVFOLYAM
1988/3. SZÁM

A NEUMANN JÁNOS SZÁMÍTÓGÉPTUDOMÁNYI TÁRSASÁG ÉS A KISZ KÖZPONTI BIZOTTSÁG LAPJA

A kiadvány a Tudományszervezési és Informatikai Intézettel együttműködve készül

A szerkesztőbizottság vezetője:
Kovács Győző

A szerkesztőség munkatársai:
Babos János (tervezőszerkesztő)
Bakos Tamás (programozástechnika)
Broczkó Péter (hírek)
Énekes Ferenc (KISZ)
Jakab Ágnes (olvasószerkesztő)
Kovács Győző (levelezés)
Krasznai Éva (diákrovat)
Lindner László (sakk)
Petróczy Judit (könyvek)
Pinke György (NJSZT, alkalmazások)
Simonyi Endre
Szebenszki Sándor
Szulyovszky Csaba
Tamásné Lakó Erika
Terebessy Ákosné
Varga András (TII, iskola-számítógép)
Vizessy Mária (μinform)

Címképünk:
Ramocsai Imri munkája

Felelős szerkesztő:
Könyves Tóth Pál

Szerkesztőség:
1027 Budapest, Fő u. 68.
Telefon: 154-250

Levélcím:
1371 Budapest
Pf. 433.

Kiadja az Ifjúsági Lap- és Könyvkiadó Vállalat

Felelős kiadó:
dr. Király G. István
igazgató

Kiadóhivatal:
1065 Budapest, Révay u. 16.
Telefon: 116-660

Terjeszti a Magyar Posta
Előfizethető a hírlapkézbesítő hivataloknál
és a Posta Hírlapelőfizetési és Lapellátási Irodáján
(1900 Budapest XIII., Lehel út 10/A.)
vagy átutalással a 215—96 162 pénzforgalmi jelzőszámra.

Megjelenik havonta
Egy szám ára 30,— Ft
Előfizetési díj:
egy évre 360,— Ft
fél évre 180,— Ft
Külföldön terjeszti a Kultúra,
1389 Budapest, Pf. 149.
és a Magyar Média
1932 Budapest, Pf. 279.
86—0253

Szikra Lapnyomda
Budapest (88—40)
Felelős vezető:
Csöndes Zoltán vezérigazgató

INDEX: 25 629
ISSN 0236-6088

Tartalom

III. Országos Mikroszámítógépes Találkozó	2
Adok—veszek—cserélek	17
Mit tud a LINGUASOFT?	20
Konzervlevelek	22
Mikrokalauz	23
Logikai szintjelző	24
Zene és tánc számítógépen	27
Egy sarokkal olcsóbb	30
Programtermékek	31
Univerzális perifériakártya	32
Mi és a számítógép, a televízió, a video és egyebek	33
Újföld	40
Olvastunk . . .	42
μINFORM	44
ISKOLA-SZÁMÍTÓGÉP	
TechnoMIR	3
HT-1080Z Commodore floppyval	4
Gyorsított rendezőprogram	5
Mire használjuk és mire ne használjuk a számítógépet?	6
DIÁKROVAT	
Programcsemegék	7
Bázisjáték HT-re	8
Megszakításvezérelt manómozgatás C64-re	8
ZX-Spectrum gyors feltöltése	9
Hangzó billentyűk Plus/4-re és C16-ra	9
Óra a keretben	10
PROGRAMOZÁSTECHNIKA	
BASIC és gépi kód	11
Z80 programok haladóknak Spectrumra és Primóra	12
Számítógépes grafika Pascalban. Sík transzformációk	15
Bináris fák	18
A képernyő balra forgatása	19
μPROGRAMOK	
Szekvenciális fájlt kezelő rutin	34
256 szín ZX-Spectrumon	35
μKLUB	
Integrált szoftver	36
Adom a magyarázatot! Ki ad magyarázatot?	38
A Mátra alján . . .	39
SAKK	45
AZ OLVASÓ ÍRJA	46
KÖNYVEK	47
HÍREK, ÉRDEKESSÉGEK	48
PONTVADÁSZAT	48

„Nincs olly bölcs a' világon, ki még igen sok hasznossal ne nevelhetné tudományit, mint viszont alig van olly tudatlan a' földkerekén, kitül egyet' s mást nem lehetne nagy haszonnal tanulni."

(Széchenyi István: Hitel)

μ '88 III. ORSZÁGOS MIKROSZÁMÍTÓGÉPES TALÁLKOZÓ

március 18–23.

Ha március, akkor Budapesti Tavaszi Fesztivál; ha Budapesti Tavaszi Fesztivál, akkor Utazás kiállítás; ha Utazás kiállítás, akkor μ'88 — mondhatnánk logikai sorba rendezve a tavaszi találkozó kapcsolatait.

Most már kezdem remélni, hogy 1986-ban hagyományt teremtettünk, amikor a Budapesti Tavaszi Fesztivál és az Utazás kiállítás vezetőivel közösen elhatároztuk, hogy ezután minden évben megrendezzük a mikroszámítógépet kedvelők és a szakma nagy tavaszi találkozóját.

Ha az 1987-es évre visszatekintünk, megállapíthatjuk, hogy számítástechnikai kiállításokból bő termést hozott. Budapesten és vidéken vállalatok, kutatóintézetek, társadalmi szervezetek szinte havonta rendeztek ilyen-olyan elektronikai, számítástechnikai, szoftver-, alkalmazói és még ki tudja, milyen kiállításokat. Ez a „termés" tulajdonképpen nem rossz, ha van elég kiállító és van elég érdeklődő. De sajnos sokszor sem az egyik, sem a másik nem jött össze. Voltam például olyan kiállításon, ahol bizony a kiállítók többen voltak, mint a közönség. Erről az egyik műegyetemi professzor mondása jut eszembe, akinek az óráit a hallgatók nem nagyon látogatták: „Egyszer még megérem, hogy többen leszek, mint a hallgatóim." Az idén ezt már számítástechnikai kiállításban megértük, és ha a tendencia folytatódik, akkor ez a kiállításinfláció súlyos problémát fog okozni a szakmának is, hiszen se a kiállító (az eladó), se a látogató (a vevő) nem fogja tudni, hogy hova menjen, ahol megfelelő információt adhat, illetve kaphat, továbbá hogy hol tehet szert értékes kapcsolatokra.

Úgy gondoljuk, hogy a μ'88 ebben a kiállításözönben is hézagpótló szerepet tölt be, és az előző évek hagyományait követve, sok kiállítót vonz, és még több látogatót segít hozzá az informatika alkalmazásainak megismeréséhez.

A μ'88 hagyományos, szinte már patinás rendezvénye az SzM SzM — a számítástechnika mindenkié, a számítástechnika mindenkiért — kiállítás, tavaly óta már vásár. A rendezőség a kiállítókat minden évben arra kéri, hogy azokat az eszközeiket és programjaikat hozzák el bemutatni, amelyeket az informatikában nem képzett emberek is könnyen használhatnak. Azt szeretnénk, ha a kiállítás látogatói ötleteket kapnának, hogy melyek azok a rendszerek, amelyeket a munkahelyükön, esetleg az otthoni környezetükben is alkalmazni tudnak, ily módon is segítve az informatika társadalmi méretű elterjedését. Az SzM SzM valószínűleg abban különbözik a többi hasonló kiállítástól, hogy itt általában a fejlesztők és alkalmazók elsősorban nem a legeslegújabb eredményeket (persze azokat is), hanem leginkább bevált és főleg széles körben alkalmazható, talán azt is mondhatjuk, hogy népszerű eszközeiket mutatják be a látogatóknak. Csak példaként említjük, hogy be szeretnénk mutatni a posta 1988-ban üzembe álló videotex rendszerét, hogy a hálózat jövendő felhasználói már jó előre megismerkedhessenek ennek az olyan sok országban népszerű rendszernek a szolgáltatásaival.

Az idén is — mint korábban — külön gondot fordítunk az iskola-számítástechnika eredményeinek bemutatására. Ami az előző alkalmakkor jól sikerült, azon most sem változtatunk; ebben az évben is pályázatot hirdettünk a legjobb oktatási programok, módszerek díjazására. A számítástechnikát alkalmazó tanárok és diákok közös fóruma a nem pedagógusok részére is nagy élmény szokott lenni és igen hasznos alkalom a tapasztalatok kicserélésére. Ezzel a lehetőséggel — reméljük — ebben az évben is jócskán élnek majd mind a diákok, mind a pedagógusok.

Az idei iskolai programunk külön színfoltja lesz a LOGO-bemutató; ez a magas szintű programozási nyelv különösen a legifjabbak számítógépes gondolkodásának fejlesztésére alkalmas. Szándékosan nem írtam programozástanítást, ti. a gyerekek ezt is megtanulják, de végül is nem ez a cél. A cél, ismétlem, az algoritmikus gondolkodásra, a problémamegoldásra való hajlam kifejlesztése. Sajnos a LOGO eddig nem volt népszerű sem az iskolákban, sem a házi számítógépes munkában, de reméljük, hogy ezen együttes erővel sikerül változtatni.

A Neumann János Számítógéptudományi Társaság keretében egyre szebben terebélyesedik a hazai szoftver, hardver és alkalmazói amatőr mozgalom. A μ'88 rendezői az idén is kötelességüknek érzik, hogy a mozgalom eredményeit, az amatőrök által kifejlesztett új eszközöket, programokat, alkalmazói rendszereket ne csak a nagyközönséggel, de a szakmával is megismertessék. Szeretnénk egy fórumot is rendezni, ahol a profi számítástechnikusok — nevezetesen gyári konstruktőrök, kutatók, mérnökök — az amatőrökkel találkoznak, és néhány, valamennyiüket érdeklő problémát megvitatnak, mint például, hogy használja-e a szakma az amatőrök eredményeit vagy segítik-e a professzionalista intézmények az amatőröket, és ha nem, akkor mit kell az együttműködés érdekében tenni. Azt hiszem, hogy egy olyan év kezdetén, amikor a műszaki haladás támogatása, a műszaki értelmiség helyzete és megbecsülése az állami politika egyik legfontosabb kérdésévé vált, különösen fontos volna ennek a problémának a megvitatása.

Annak ellenére, hogy többször is megpróbáltuk, nem tudtunk lényegesen előbbre lépni a hazai μklubok összefogása és tevékenységük fellendítése érdekében. Sok helyen, például művelődési házakban, iskolákban, társadalmi szervezetek keretében igen virágzó klubélet folyik, máshol semmi. Talán egy, a klubokról szóló újabb tapasztalatcsere segítene a mozgalomnak új célokat és mozgósító feladatokat adni.

Nem felejtjük ki a programból a számítástechnika vagy az informatika népszerűbb alkalmazásait sem, mint például a játékokat. Ezek közül elsősorban a sakkot és a bridzset játszó gépekkel rendezünk versenyeket, sőt a tavaly nagyon jól sikerült szimultánt is megrendezzük, egy kicsit más formában.

Meghirdettünk egy karikatúrakiállítást „A gép is ember" címen, ahová részben a hagyományos módszereket alkalmazó karikaturisták, részben a számítógéppel alkotó művészek pályaműveit várjuk.

Lesz ismét — talán jobb propagandával megszervezett — számítógépes hangverseny; ha sikerül, akkor komolyzenei és könnyűzenei is, neves művészek részvételével, valószínűleg nem a kiállítás területén, hanem valamelyik jó akusztikájú hangversenyteremben.

Ismét rendezünk családi versenyt, amelylyel azt szeretnénk bemutatni, hogy a drága pénzen vásárolt otthoni számítógépeket a játékon kívül mennyi másra is lehet alkalmazni.

Az Országos Műszaki Múzeum ismét új kiállításra készül. Ezúttal nemcsak öreg számítástechnikai eszközöket hoz, de az adatátvitel, az információrögzítés historikus emlékeit is láthatják a látogatók.

Írásom elején említettem, hogy a μ'88 már nemcsak kiállítás, nemcsak találkozó, de vásár is. Tavaly is voltak üzletek, amelyek tulajdonosai főleg könyvekből, számítógépes médiumokból, de perifériákból és számítógépekből is jó forgalmat csináltak. Így akik kijöttek a μ'88-ra, általában nem mentek haza üres kézzel, mert olyasmit is megvehettek, amit egyébként csak hosszas utánjárás után tudtak volna megkapni. Reméljük, az idén is lesz olyan vásárfia, amit szívesen vesznek meg a látogatók a μ'88 üzleteiben.

Egy folyóirat cikkterjedelme véges. Nyilvánvaló, hogy a μ'88 minden eseményéről nem lehet írni; csak egy kis ízelítőre vállalkoztam. Azt tanácsolom, hogy jöjjenek ki a μ'88-ra, ahol a rendezőbizottság nevében szíves örömmel várja a kedves olvasókat

KOVÁCS GYŐZŐ

TechnoMIR

A próbapanel

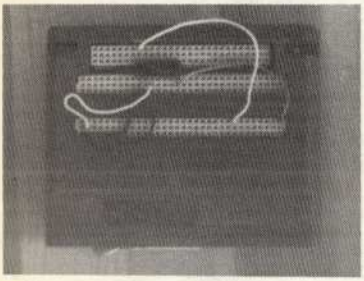
Másfél DINT és másfél DOUT egyszerre!

A Proto Board (PB) modul próbapanelnak is nevezhető, mert a tetején elhelyezett három csatlakozósávon különböző elektronikus áramkörök valósíthatók meg. Az alkatrészek (IC, LED, tranzisztor stb.) a szabványos méretű csatlakozókba dugaszolhatók, az összeköttetéseket pedig vékony, merev, szigetelt vezetékkel, ún. gyengeáramú szerelőhuzallal lehet kialakítani.

A modul fő előnye, hogy a megépített áramkör vagy a problémás alkatrész állapotai — ha azok a TTL szinteknek megfelelnek — számítógép segítségével, programból vizsgálhatók. A programozhatóságot a csatlakozósávra kivezetett, háromszor 8 bites univerzális I/O felület biztosítja.

A csatlakozók kialakítását a modulról készült *1. kép*, illetve a *3. ábra* mutatja. Jól

2. kép



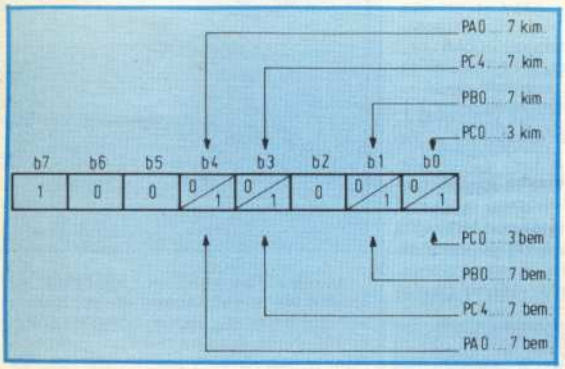
1. kép

látható, hogy két sáv üres; itt célszerű az áramköri elemeket elhelyezni. A harmadik csatlakozósávra vannak kivezetve a ki- és bemeneti pontok (az I/O regiszterek pontjai), valamint a gerinchálózat néhány vonala.

Felhívjuk a figyelmet arra, hogy a +12 V, −12 V tápfeszültséghez csak a POW (tápegység) modul alkalmazásával juthatunk, a számítógép csak a +5 V-os táplálást biztosítja. Külső tápegységet akkor is kell alkalmazni, ha a kialakított áramkör és a PB modul áramfelvétele együttesen meghaladja a számítógép megengedett terhelhetőségét.

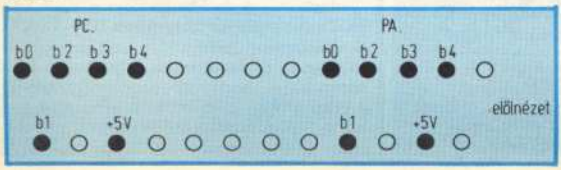
A PB modul lelke egy 8255 típusú IC. Ez az LSI áramkör az Intel MCS-80 családjába tartozik, és több különböző üzemmódban párhuzamos adatforgalmat képes lebonyolítani. A modulban a legegyszerűbb, ún. Ø üzemmódban dolgozik.

A mi szempontunkból a felépítését úgy képzeljük el, hogy három darab, egyenként 8 bites programozható regisztert (PA, PB és PC azaz Port A, Port B és Port C) tartalmaz. A programozhatóság itt azt jelenti, hogy beállítható, melyik regiszter legyen kimenet és melyik bemenet. Az utolsó, PC re-

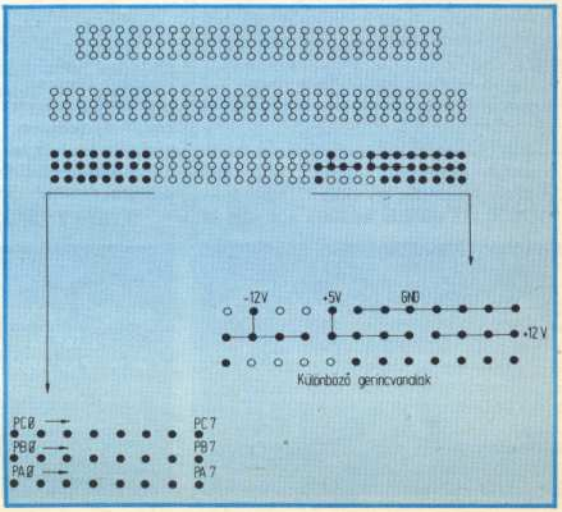


1. ábra

2. ábra



3. ábra



giszter 4 bitenként programozható, tehát lehet olyan állapota, amikor az alsó 4 bit (PCØ-PC3) például bemenet, de a felső 4 bit (PC4—PC7) kimenet.

Az egyes regiszterek állapotát hordozó információ tárolására egy kontrollregisztert helyeztek el az áramkörben; a PB modul használatát ennek feltöltésével kell kezdeni. A regiszter HT címe 43, a megfelelő utasítás tehát az OUT 43,X. Egyéb számítógépeknél a megfelelő cím írandó (lásd a Magazin 1987/12. számában!).

Az X szám, azaz a kontroll szó az *1. ábra* alapján képezhető. A PA, PB és PC regiszterek HT címei rendre 40, 41 és 42. A kimenetek aktív 5 V-os kimenetek, a bemenetek nem tároló bemenetek.

Programozási példát nem hozunk, mert a PB nem más mint (beállítástól függően) egy DINT és egy DOUT együtt, amit pedig remélhetően mindenki kiválóan programoz már. A fényképen látható összeállítási példa egy 7400 logikai IC egyik NAND kapujának vizsgálatára alkalmas.

Egy illesztő interfész

A JOY (joystick-illesztő) modul két szabványos botkormány fogadására alkalmas. Elsősorban természetesen azoknál a számítógépeknél van jelentősége, amelyekbe nincs ilyen áramkör beépítve, tehát a HT, a Spectrum vagy a Primo esetében. Ez utóbbi tartalmaz ugyan botkormányillesztőt, de hatpólusú DIN aljzatához a legelterjedtebb típusok nem csatlakoztathatók.

A JOY modul segítségével ugyanazt a botkormányt használhatjuk, mint a Commodore gépekhez. Sőt, ha a modult például C64-hez kötjük, a csatlakoztatható botkormányok száma és így a számítógéppel közvetlenül kapcsolatot tartó személyek száma négy lehet.

Egy lehetséges összeállítást mutat a *2. kép*. A csatlakozó bekötése a *2. ábrán* látható.

A felhasználói program elején a modul kontrollregiszterébe 155-öt kell írni. A kontrollregiszter HT címe 47, tehát HT-n OUT 47,155 utasítás hatására a modul képes a botkormány jeleit fogadni és továbbítani a számítógépnek. Az információt a PA bemenet esetén a 44 HT címről nyerjük — INP(44) —, a PC bemenetet használva 46 a HT cím. Egy programon belül természetesen mindkettő használható.

```
    ****************************
    *                          *
    *  RAJZPROGRAM  PRIMÓRA    *
    *                          *
    ****************************

 10 CLS
    * KERET MEGRAJZOLÁSA     *
 20 FOR I=0 TO 255
 30 SET(I,0) : SET(I,191)
 40 NEXT I
 50 FOR I=0 TO 191
 60 SET(0,I) : SET(255,I)
 70 NEXT I
 75 X=128 : Y=96
    * CONTROLL SZÓ BETÖLTÉSE *
 80 OUT 192+47,155
    * BILLENTYŰZET VIZSGÁLATA *
 90 A$=INKEY$
100 IF A$="" THEN 110 ELSE 10

    ****************************
    * JOY MODUL LEKÉRDEZÉSE    *
    ****************************

110 A=INP( 192+47 )
120 IF A AND 2 THEN 200
    * JOYSTICK LEKÉRDEZÉS RAJZOLÁSRA *
130 IF A=1 THEN Y=Y+1
140 IF A=4 THEN Y=Y-1
150 IF A=8 THEN X=X-1
160 IF A=16 THEN X=X+1
170 SET (X,Y) : GOTO 90
    * JOYSTICK LEKÉRDEZÉSE TÖRLÉSRE *
200 IF A=3 THEN Y=Y+1
210 IF A=6 THEN Y=Y-1
220 IF A=10 THEN X=X-1
230 IF A=18 THEN X=X+1
240 RESET (X,Y) : GOTO 90
```

A megfelelő címről beolvasott adat a botkormány állapotát bitenként tartalmazza a következő módon:

Ø bit — fel
1. bit — tűz
2. bit — le
3. bit — balra
4. bit — jobbra

Tehát a 44, illetve 46 címről beolvasott 1; 2; 4; 8; 16 adat alapján eldönthető a botkormány állapota.

A közölt program Primóra készült egyszerű rajzprogram, elsősorban a modul használatának gyakorlásához. Minimális változtatás után érdemes HT gépen is kipróbálni.

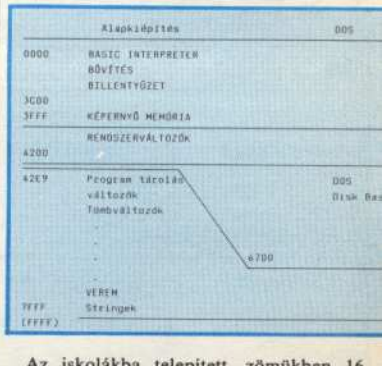
KIRÁLY LÁSZLÓ—ALBU LÁSZLÓ

HT—1080

A HT—1080Z számítógép ROM programja egy sor, alapkiépítésben nem használt kulcsszót tartalmaz. Az alapgépben ezeknek a kulcsszavaknak ugrási címei az SN hibára mutatnak. Teljes kiépítésénél, csatlakoztatott floppyval például a TRSDOS NEWDOS 2.1 verziója jelentkezik be. Ilyenkor élnek a *táblázatban* felsorolt DOS kulcsszavak és rutinjaik. (Vö. Sztrókay: A Z80 Assembler. Műszaki Könyvkiadó, 1985. Felföldi—Lukács: A HT—1080Z iskolaszámítógép fix tárolója. Ipari Informatikai Központ, 1985.)

CVI	5E46	GET	627C
FN	558E	PUT	627B
CVS	5E49	CLOSE	606F
DEF	5655	LOAD	5F78
CVD	5E4C	MERGE	6008
EOF	61EB	NAME	6346
LOC	6231	KILL	63C0
LOF	6242	&	58B7
MKI$	5E20	LSET	60E6
MKS$	5E30	RSET	60E5
MKD$	5E33	INSTR	582F
CMD	56C4	SAVE	6044
TIME$	5714	LINE	5756
OPEN	6349	USR	5679
FIELD	60AB		

A végrehajtó rutinok címeiből látható, hogy a rendszer a memóriát is átszervezi.



Az iskolákba telepített, zömükben 16 kbájtos RAM-mal üzemelő gépeknél ilyen kiépítés esetén alig marad szabad memóriaterület. Pedig de jó lenne a HT gépeknél is floppyt használni! A hajlékonylemezes tárolónak a kazettás tárolóval szembeni előnyeit úgy vélem, nem kell ecsetelni.

Idestova egy éve használom a Real-Team Gmk sorosbusz illesztőjét a HT—1080Z és Commodore perifériák összekapcsolására (lásd az *ábrát*). Az összeállításban a HT jelentéktelen korlátokkal ugyanúgy képes kezelni a VC—1541-et, mint azt a C64 teszi. A bővítő csatlakoztatása után a HT értelmezi és lemezmeghajtó egység (és/vagy

Minden hétfőn 17-től 19 óráig

a Mikroszámítógép Magazin munkatársai és felkért szakértők válaszolnak az olvasók kérdéseire a szerkesztőségben: Budapest II., Fő u. 68. I. em. 109. vagy a 154-090 és a 154-250-es telefonon.

Minden kedden 17-től 20 óráig
ENTERPRISE klub
a VSZM Közösségi Házban
(Bp. XI., Fehérvári út 120.)
Klubvezető: Romvári Gábor
Telefon: 450-950/473

Commodore floppyval

nyomtató) kezelésére felhasználja a LOAD, LOAD? (=VERIFY), SAVE, OPEN, CLOSE, CMD, PRINT*(=PRINT#), GET*, INPUT*, PUT* kulcsszavakat.

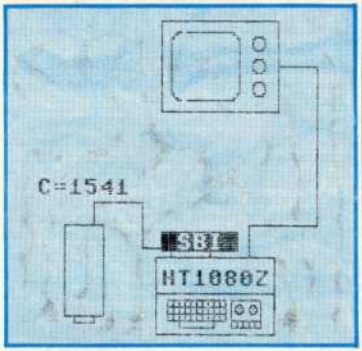
A HT ROM-ban lévő, TRSDOS részére fenntartott — és ezzel az összeállítással nem értelmezett — kulcsszavak saját rutinok meghívására továbbra is rendelkezésre állnak.

Az illesztőegység a HT RAM-jából csak a 403EH-407FH területet és az I/O puffert használja fel. Az illesztést és az új kulcsszavak végrehajtását az interfész saját 2 kbájtos rendszerprogramja biztosítja, mely párhuzamosan él a HT ROM-jával.

Minthogy a HT a VC—1541-et kezeli, a Commodore gépeknél alkalmazott lemezformátumot hozza létre, így ugyanúgy alkalmazhatók a parancscsatornán a közvetlen elérésű üzemmód DOS parancsai is. Ez a lehetőség új távlatot nyit a HT programozásában és az adatkezelésben. Az itt közölt program lemezfoglaltsági térképet rajzol a képernyőre.

Az illesztő használatának nagy előnye, hogy nagymértékben csökkenti a HT iskolai oktatásban való alkalmazásának „rizikófaktorát". A saját készítésű programokon kívül a TII oktatóprogram-jegyzékében közölt mintegy 300 program is az iskolák rendelkezésére áll. Az órákon való felhasználás fő megnehezítője, a kazettás tárolás miatt azonban e programokat viszonylag kevéssé használják.

A floppy alkalmazása az 500 bit/s átviteli sebességet 3200 bit/s-ra emeli; ugyanakkor tudjuk, hogy a programbeolvasás időszükségletét nemcsak a LOAD parancs tényleges végrehajtásából adódik. Így bizvást állítom, hogy a program betöltési ideje és megbízhatósága nagyságrenddel gyorsabb, jobb, mint korábban, a „kazettás világban" volt.



A BASIC programok minden további nélkül átmásolhatók kazettáról lemezre. A gépi kódú (játék) programok kazettáról lemezre töltésének több módja van. Az egyik megoldás a következő lépésekből áll:

1. A gépi kódú program kezdő-, vég- és indítási címének meghatározása

2. A gépi kódú program kazettáról való betöltése

3. Indítás helyett RESET vagy BREAK (öninduló programnál RESET)

4. A BASIC-terület programcimmutatóinak átírása a gépi kódú program megfelelő címeire

5. BASIC programként a program lemezre vétele: SAVE „gépi", 8

Az így kimentett program LOAD „gépi", 8,1 paranccsal betölthető és SYSTEM / KEZDŐCÍM begépelésével indítható.

(Az egyszerűbb, „nagyüzemi" megoldáshoz a gmk által gyártott, több gép összekapcsolását biztosító Minet interfészt is alkalmazni kell.)

A közkézen forgó gépi kódú HT játék- és felhasználói programok zöme ezzel a módszerrel lemezre vihető. Vannak azonban olyan kazettás gépi kódú programok is, amelyek használják a DOS memóriaterületét. Az ilyen programok betöltése a rendszer összeomlásához vezet. A sorosbusz interfész használata során a kulcsszavak ugrási címeinek (41734—41E2H), a DOS munkaterületnek (403EH—407FH) felülírásán, valamint a SYSTEM/Ø alkalmazásán kívül más módszerrel még nem sikerült kiakasztani.

Nagy kár, hogy nem korábban és nem nagyobb darabszámban került az interfész alkalmazásba — de lehet, hogy ebben nem is ő a hibás.

SZ. LUKÁCS JÁNOS

```
10 CLEAR1000:DEFINT A-Y
20 CLS:PRINTTAB(5)"LEMEZ FOGLALTSAG"
30 FOR I=10T045STEP2:SET(10,I):NEXT
40 FOR I=11T036STEP2:SET( I,9):NEXT
50 FOR I=5T09          :SET(I,27):NEXT
60 FOR I=10T026:FORT=32T035:SET(T,I):NEXT:NEXT
70 FOR I=27T033:FORT=30T035:SET(T,I):NEXT:NEXT
80 FOR I=34T039:FORT=29T035:SET(T,I):NEXT:NEXT
90 FOR I=40T044:FORT=28T035:SET(T,I):NEXT:NEXT
100 PRINT@320,"SRV.";:PRINT@340,"SZEKTOR:";
110 OPEN15,8,15,"I0":OPEN2,8,2,"#"
120 PRINT#15,"B-R";2;0;18;0
130 PRINT#15,"B-P";2;144;N#=""
140 FOR I=1T020:GET#2,X#
150 IF X#=CHR#(13) OR X#=CHR#(160) THEN X#=" "
160 IFX#=""THENGOTO170
170 N#=N#+X#:NEXT:CLOSE2:CLOSE15
180 PRINT@64,"NEV/ID:";N#;
190 OPEN15,8,15,"I0":OPEN2,8,2,"#"
200 PRINT#15,"B-R";2;0;18;0
210 S=0:FORA=4T0140STEP4:PRINT#15,"B-P";2;A
220 PRINT@ 384,A/4;
230 FOR I=1T04:GET#2,X#:IFX#=""THENX#=CHR#(0)
240 W=ASC(X#):IFI=1ANDA<>72,S=S+W
250 IFI<>1THENGOSUB290
260 NEXT:NEXT:CLOSE2:CLOSE15
270 PRINT@128,"SZABAD:";S,"BLOKK",
280 IF INKEY#=""THEN280 ELSE END
290 F=9+A/4:V=-6+I*8:O=128
300 Z=W/O:V=V+1:IFZ<1THENGOSUB380
310 C=V:IF(A/4<18ANDV>31)THENC=31
320 IF(A/4>17ANDA/4<25ANDV>29)THENC=29
330 IF(A/4<31ANDA/4>24ANDV>28)THENC=28
340 IF(A/4>30ANDV>26)THENC=27
350 PRINT@406,C-10;
360 W=W-INT(Z)*O:O=O/2:IFO>=1THEN300
370 RETURN
380 IFPOINT(V,F)=0THENSET(V,F):RETURN
390 RESET(C+1,F):O=0:RETURN
```

PRIMO

Gyorsított rendezőprogram

A Mikroszámítógép Magazin 1987/8. számában az Integrált szoftver című cikkben olvastam, hogy hosszú szövegrészek mozgatása esetén néha úgy tűnik, mintha a számítógép „lefagyott" volna.

Ezt a jelenséget tapasztaltam a Primónál is nagyobb méretű szöveges adathalmazok rendezése esetén. Ilyenkor hajlamosak vagyunk a feladat megoldását kizárólag a gépi kódú programban keresni. A BASIC nyelv gépi lehetőségeit felhasználva rendezőprogramunk gyorsítható, ha rendezés közben a szöveges adatokat nem mozgatjuk. A program működéséhez tudni kell, hogy a Primo a következő módon tárolja a szöveges változókat:

```
10000 REM  GYORSITOTT RENDEZO ........
10010 N=4
10011 AI=0 : AJ=0 : I=0 : J=0
10012 A0=0 : A1=0 : A2=0
10160 DIM A$(4)
10170 A$(1)="FAL"
10180 A$(2)="ASZTAL"
10190 A$(3)="TV"
10200 A$(4)="PAPIR"
10410 REM RENDEZES.....................
10420 FOR I=1 TO N
10422 FOR J=I TO N : PRINT N-I;
10430 IF A$(J)>=A$(I) THEN 10470
10440 AJ=VARPTR(A$(J))
10441 AI=VARPTR(A$(I))
10442 A0=PEEK(AJ)
10443 A1=PEEK(AJ+1)
10444 A2=PEEK(AJ+2)
10447 POKEAJ,PEEK(AI)
10448 POKEAJ+1,PEEK(AI+1)
10449 POKEAJ+2,PEEK(AI+2)
10450 POKEAI,A0
10451 POKEAI+1,A1
10452 POKEAI+2,A2
10470 NEXT J
10480 NEXT I
10490 REM KIIRATAS.....................
10495 CLS
10500 FOR I=1 TO N
10510 PRINT A$(I)
10520 NEXT I
```

1. a változó VARPTR függvénnyel visszakereshető címén a változó hosszát karakterekben,
2. a következő és a következő utáni címen azt a címet, ahol a szöveget ténylegesen tárolja.

A rendezés elvégzéséhez megírunk egy „normál" rendezőprogramot, de a szöveges adatok cseréjét nem végezzük el, hanem a két kicserélendő változó tárolási helyét megkeressük a VARPTR függvénnyel, és az ezeken a címeken és az utánuk következő két címen lévő értékeket egymással felcseréljük. A mellékelt program segítségével ellenőrizhetjük a leírtakat.

SOMOGYI GYÖRGY

Mire használjuk és mire ne használjuk a számítógépet?

Az NJSZT diplomaterv-pályázatán első díjat nyert dolgozat egyik fejezete a számítógépek középiskolai alkalmazási lehetőségeivel foglalkozik. A pályamunkának ezt a részét közöljük, némi rövidítéssel.

Számítások végzése

Jól felhasználható a számítógép mérések kiértékelésénél, ahol a grafikus megjelenítés szemléletesebbé teszi a megoldást. Szerepe lehet bonyolult képletrendszer kiszámításában vagy olyan feladat elvégzésénél, amely ha egyszerű is, de sokszor kell ismételni (például a statisztikus fizikában, egyenletek, egyenletrendszerek megoldásánál). Nem szabad alkalmazni a számítógépet, ha a számítás menete az érdekes, vagy akkor, ha egy zsebszámológép is el tudja végezni a számítást — azaz ne lőjünk verébre ágyúval.

Oktatási segédprogramok

Olyan feladatok megoldását segíthetik, ahol — már vagy még — nem a tevékenység elvégzése a lényeges. Ide tartoznak a versek ritmikai és hangtani elemzését segítő programok, a billentyűzetről lejátszott zenéhez kottarajzolást végző programok, a függvényábrázolást, függvénytranszformációt végző programok stb. Mindezek hasznos eszközök ugyan, de önmagukban nem indokolják a számítógépek széles körű alkalmazását az iskolában.

rendszerben oldhatók meg, ahol lehetőség van nagy mennyiségű adat tárolására és visszakeresésére.

A programozott oktatás számítógépes segítése

E programok pedagógiai haszna erősen megkérdőjelezhető, ráadásul a jelenlegi számítógépes kiépítettség mellett még egy könyv lehetőségeit sem érik el. Magyarországon nem gyakoriak az ilyen programok.

Logikai játékok

Hasznosak lehetnek nemcsak a számítógéppel való ismerkedés idején, de a programozás tanulásakor is. Értelmes játékprogramok írása a logikus gondolkodást is fejleszti.

Számítógépes szimuláció

A számítógép nem helyettesítheti a kísérletet. Akkor jut szerephez, ha a kísérletezés valamilyen okból akadályba ütközik, mert a jelenség túlságosan gyors (az atomban lejátszódó folyamatok), lassú (csillagászati, geológiai jelenségek), drága (atomreaktor helyes működéséhez szükséges paraméterek beállítása), veszélyes (mérgező anyagok, láncreakció esete), bonyolult (az agy); a kísérletet túl sokszor kell elvégezni (kockadobás), vagy etikai akadályokba ütközik (emberrel végzett kísérlet) stb.

A számítógépnek nemcsak hiánypótló szerepe lehet, hanem intelligens kísérlettervezőként is megállja a helyét. Segítségével gyorsan és hatékonyan megvalaszthatók az elvégzendő kísérletek paraméterei. A szimulációs modell elkészítésére kétféle lehetőség adódik. Amikor elemi folyamatokat modellezünk, a kezdeti paraméterek megadása után a jelenség szimulálása az események véletlenszerű kiválasztásával történik. Ekkor a számítógépes modell igen közel áll a természethez. A másik lehetőség, hogy a szimuláció a természeti törvények már felállított egyenletei alapján modellezi a jelenséget. Az ilyen megoldás akkor hasznos, ha nagyon sok folyamat együttes hatását kell vizsgálnunk, amelynek áttekintése és a kezdeti paramétereitől függő eredmény megadása nagyon nehéz lenne. Ilyen például egy atomreaktor folyamatainak modellezése, ami azért is kínálkozik az egyetlen lehetőségnek, mert egy valódi reaktor paramétereinek változtatása igen veszélyes.

A szimulációs megismerési folyamatnak iskolai alkalmazás során a következő lépései képzelhetők el:

— foglalkozzunk a valós rendszerrel, ismerjük meg a törvényeit, jelenségeit,

— beszéljük meg a vizsgálni kívánt jelenséget; ha lehet, figyeljük meg,

— szakköri foglalkozáson készítsük el a modelljét,

— számítástechnikai szakkörön készítsük el a számítógépes modellt és a programot,

— használjuk a számítógépes modellt, keressünk vele érdekes paramétereket, vizsgáljuk meg a kapott eredményeket, próbáljuk kitalálni az összefüggéseket,

— ha lehet, kísérletezzünk valós rendszerrel és használjuk fel a szimulációs eredményeket,

— ha kísérleteztünk, vessük össze a kísérletekkel és a szimulációval kapott eredményeket,

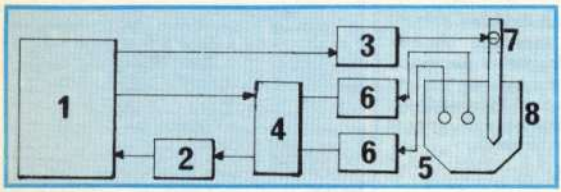
— értelmezzük a kapott eredményeket,

— ha szükséges, ennek alapján módosítsuk a modellt és a számítógépes modellt.

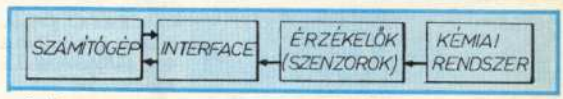
A szimuláció témakörében a számítógépes feladatok száma határtalan, így nagyon sokféle program készülhet. Ezért vált a szimuláció a számítógépek felhasználásának egyik legfontosabb területévé.

Számítógépes mérés és vezérlés

A számítógépek használatának ugyancsak fontos területe a számítógépes mérés és vezérlés. Az igazán magas szintű mérések azok, amelyeket számítógép vezérel. Ezekben a számítógép különböző eszközök segítségével, a program alapján beavatkozik a mérés menetébe, és például valamihez hozzáönt egy reagenst, vagy elzár egy csapot.



1. ábra



2. ábra

Egy számítógéppel vezérelt kísérlet rendszerfelépítését mutatja az *1. ábra*, amelynek értelmezése a következő. A folyamatok által szolgáltatott jelek általában nem digitálisak, hanem rendszerint analógok, ezért a számítógép (1) és a külvilág közvetlen csatlakozása egy analóg-digitál (2), illetve egy digitál-analóg (3) átalakítást kíván. A több mérőhelyről érkező jelek egymás utáni befogadását a multiplexer (4) teszi lehetővé. A kémiai és fizikai adatok felfogására és a szükséges elektromos jellé alakítására érzékelők kellenek (5), amelyek jele először rendszerint egy jelfeldolgozó (6) mérőműszerbe jut (például Wheatstone-híd). A mérés vezérlése a megfelelő kiegészítő eszközök (bürettacsap, termosztát stb.) működtetésével történik (7). Természetesen a főként hagyományos laboratóriumi eszközökből álló reakciótér (8) jelenléte is szükséges.

Ilyen típusú mérést ma még nem végezhetünk az iskolában, mert a beavatkozáshoz szükséges eszközök nagyon drágák. Ezért be kell érnünk azzal, hogy a beavatkozást a kísérletező végzi, a számítógép pedig csak mér és figyelmeztet; jelzi, mi a következő teendőnk. Az iskolai alkalmazást szem előtt tartva talán nem is túl jó, ha egy mérés teljesen automatizált. Nagyobb probléma, hogy az ilyen, „félig automatizált" méréshez szükséges érzékelők, illesztők sem állnak korlátlanul rendelkezésünkre. A *2. ábra* a számítógéppel csatolt mérés felépítésének vázlatát tartalmazza, amely tájékoztat arról, hogy milyen egységek kellenek egy ilyen kísérlethez.

Az Országos Pedagógiai Intézet számára 1985-ben készült felmérés szerint az iskolák 69 százalékában van az ilyen mérésekhez megfelelő számítógép, de már csak alig 10—15 százalékuk tudja biztosítani a legfontosabb érzékelőket és mérőműszereket is. Az illesztők hiányán enyhít a TechnoMIR-készlet megjelenése, de a probléma végleges megoldása itt is várat magára.

GÖRBICS LÍVIA

Tesztek végzése és kiértékelése

Ez a számítógépek legsematikusabb felhasználási területe. A jó programok is alig nyújtanak többet, mint a hagyományos tesztek, de a legtöbb esetben még ezt sem teszik, csupán a szöveg megjelenítését és az értékelést végzik. Sokszor maguk a tesztkérdések is sok kívánnivalót hagynak maguk után. (Számomra az is kétséges, hogy helyes-e az iskolai számonkérésben teszteket alkalmazni, különösen akkor, ha figyelembe vesszük, hogy megoldásuk eredményessége a matematikai képességekkel korrelál.)

Egyéni gyakoroltatás

Elvileg lehetséges ilyen jellegű felhasználás is, például idegen szavak vagy idegen nyelvű ragozás gyakorlására. Ebben az alkalmazásban azonban különösen sok buktatója van egy program elkészítésének, és egy rossz program — például a válaszok túl merev értékelésével — többet árthat, mint használ. Elképzelhető azonban olyan alkalmazás is, hogy a pedagógus személyre szabottan készít kisebb-nagyobb programokat ismétlésre, otthoni tanulásra, olyan gyerekek számára, akik versenyre készülnek vagy hosszabb ideig betegek, és ezért szükséges, hogy egyénileg gyakoroljanak, megfigyeljenek bizonyos törvényszerűségeket, vagy éppen számoljanak.

Információtároló és -visszakereső programok

Ilyen például egy iskolai könyvtár könyvállományát kezelő vagy egy növényhatározót létrehozó program. Az iskolai alkalmazás kerékkötője, ha nincs megfelelő háttértároló; ezek a feladatok ugyanis csak lemezegységet tartalmazó

Programcsemegék BEOS módra

Optimizer

A program azoknak nyújt segítséget akik szeretnék minél jobban kihasználni a mágneslemez kapacitását. Igyekszik úgy összeválogatni a tárolandó programokat (méreteik alapján), hogy a lehető legkevesebb szabad hely (blocks free) maradjon a lemezen, a tűréshatár 'szem előtt' tartásával. Az üres szektorok száma ezen értéken belül lesz. Tehát ha TURESHATÁR-nak 8-t adunk, a programot arra kényszerítjük, hogy a rendelkezésre álló blokkméretekkel addig sakkozzon vagyis válogasson közülük, míg összegükként 664-et nem kap. A futási időre való tekintettel, ezt rögtön kiírja, majd csökkentve a készletet tovább keresgél.

A programlista PRINT utasításaiban szereplő kisbetűs rövidítések:

sc = SHIFT+CLR HOME bill.
ion = CTRL+RVS ON bill.
iof = CTRL+RVS OFF bill.

```
100 PRINT"sc": PRINT: PRINT TAB(12)"ion DISK OPTIMIZER iof"
102 PRINT: PRINT SPC(7)"FILE-OK LEMEZEN TAROLASAT"
104 PRINT SPC(9)"OPTIMALIZALO PROGRAM."
106 PRINT: PRINT
110 INPUT"PROGRAMOK SZAMA";A
120 IF A<2 OR A>1000 THEN 110
130 DIM T(A), TT(A), V(A): AA=A
150 FOR I=1 TO A
154 :   PRINT
156 :   FOR J=1 TO A
160 :     PRINT J;"BLOKKMERET ";
162 :     C=PEEK(211): PRINT T(J);
164 :     POKE 211,(C-1): INPUT B
166 :     IF B<1 OR B>664 THEN 160
170 :     T(J)=B: TT(J)=B: B=0
180 :   NEXT J: PRINT
184 :   IF I=1 THEN PRINT"ELLENORZES!"
190 NEXT I
192 INPUT"PRINTER VAN";V$: PRINT
194 IF V$="" THEN 192
196 IF V$="I" OR V$="IGEN" THEN OPEN 4,4: Q=1
200 INPUT"TURESHATAR";B: PRINT
210 IF B<0 OR B>664 THEN 200
220 REM------BACKTRACK ALGORITMUS--------
230 GOSUB 400: S=0: P=1: Z=0
240 FOR I=P TO A
250 :   S=S+T(I)
260 :   IF S>664 THEN S=S-T(I): GOTO 290
270 :   Z=Z+1: V(Z)=I
280 :   IF 664-S <= B THEN GOSUB 500: GOTO 230
290 NEXT I
300 S=S-T(V(Z)): P=V(Z)+1: Z=Z-1
310 IF Z<0 THEN 600
320 IF P<=A THEN 240
330 GOTO 300
400 REM--------SZELEKTIV RENDEZES--------
410 FOR J=1 TO A-1
420 :   FOR K=J+1 TO A
430 :     IF T(K) > T(J) THEN M=T(J): T(J)=T(K): T(K)=M
440 :   NEXT K
450 NEXT J
460 RETURN
500 REM----------EREDMENY KIIRAS----------
505 IF Q THEN PRINT#4,S;"BLOKK:";
510 PRINT S;"BLOKK:";
520 FOR J=2 TO 1 STEP -1
525 :   IF Q THEN PRINT#4,T(V(J));
530 :   PRINT T(V(J)); : T(V(J))=T(A): A=A-1
540 NEXT J: PRINT: IF Q THEN PRINT#4
550 IF A > 1 THEN RETURN
600 REM-----------------------------------
605 IF A=AA THEN PRINT"ion NINCS MEGOLDAS iof"
610 IF Q THEN PRINT#4,"   MARADEK:";
615 PRINT: PRINT"   MARADEK:";
620 FOR J=1 TO A
625 :   IF Q THEN PRINT#4,T(J);
630 :   PRINT T(J);
640 NEXT J: PRINT: PRINT
645 IF Q THEN PRINT#4: PRINT#4
650 INPUT"MEGFELEL";V$: PRINT
660 IF V$="I" OR V$="IGEN" THEN CLOSE4: END
670 A=AA
680 FOR I=1 TO A: T(I)=TT(I): NEXT I: GOTO 200
```

Örökélet POKE-ok

BLACK WYCHE	POKE 4248,44
COMMANDO	POKE 2409,44
CRACKERS REVENGE	POKE 32170,44
FIRE ANT	POKE 16696,44
JET SET WILLY	POKE 14711,234 POKE 14712,234 SYS 8192
MUTANT MONTY	POKE 19231,44
ROCKET ROGER	POKE 21233,238 POKE 21237,69 POKE 21238,7 SYS 5600
SON OF BLAGGER	POKE 6370,234 SYS 8035
SPELUNKER	LOAD "UUU",8,1 POKE 19125,234 POKE 19126,234 SYS 9838
THUNDERBIRDS	POKE 17669,44 POKE 25404,44
TROLLIE WALLIE	RESET után: POKE 7711,44 SYS 51712

Folyt. köv.

Készítette:
HÁMÁN KATÓ
Közgazdasági Szki
MHSZ klubja

Bázisjáték HT-re

A játékban űrhajónk meteoritokkal csatározik. Úgy kell masinánkat irányítani, hogy az össze ne ütközzön a kóbor hulló csillagokkal — ezek egyébként a képernyőn Ø-val jelzettek —, ugyanakkor minél több ellenséges objektumot (X) semmisítsen meg. Az űrhajó a talaj felett repdes, s közben változik az attól mért magassága. A játék izgalmát fokozandó, a képernyő balra tolódik.

A figurákat a fel- és lefelé mutató nyilakkal mozgatjuk, bombázni pedig a szóközzel tudunk. A gép a képernyőeltolásért 1, a sikeres találatért 30 pontot „fizet".

Ha az űrhajó meteoritba ütközik vagy a talajba fúródik, illetve ha elhagyja a képernyőt, akkor vége a játéknak. Cél, hogy minél hosszabb idő alatt, minél több pontot gyűjtsünk össze.

BACZÓ TAMÁS

```
 3 REM #  READY? 29999  #
 10 CLS:RANDOM:CLEAR 500
 20 DEFINT X,G,Z,I-N
 30 X=462:G=14400:Z=16159
 40 B$=" "+CHR$(152)+CHR$(191)+CHR$(164)
 50 C$=STRING$(32,191)
 55 F=15361
 60 POKE 32763,4:Z1=INT(Z/256)
 70 POKE 32761,Z1:POKE 32760,Z-256*Z1
 80 POKE 16526,48:POKE 16527,117
 82 IF PEEK(32767)=10,READ F$,F$,F$,F$:GOTO 200
 83 POKE 32767,10
 85 PRINT"KAPCSOLJ AT FEL KEPERNYORE !"
 90 FORI=1 TO 4:READ A$
 100 FORJ=1 TO LEN(A$) STEP 2
 110 D$=MID$(A$,J,1):E$=MID$(A$,J+1,1)
 120 IFASC(D$)>64,D=(ASC(D$)-55) ELSE D=ASC(D$)-48
 130 IFASC(E$)>64,E=(ASC(E$)-55) ELSE E=ASC(E$)-48
 140 POKE 30000+H,D*16+E:H=H+1
 150 NEXT J,I:CLS:GOTO1000
 200 FORI=0 TO 13:READ D
 210 OUT 31,I:OUT 30,D:NEXT
 220 CLS:PRINT@0,STRING$(32,131);:PRINT@X,B$;:E=7
 230 FORI=768 TO 960 STEP 64:PRINT@I,C$;:NEXT
 299 REM *** FOPROGRAM ***
 300 IFPEEK(G)=8,GOSUB1200:GOTO 325
 310 IFPEEK(G)=16,GOSUB1300:GOTO 325
 320 IFPEEK(G)=128 ANDM=0,GOSUB1500:GOTO 350
 325 IFM=1,GOSUB1510 ELSE FORI=1 TO20:NEXT
 350 IFL/3=INT(L/3),D=RND(9):PRINT@D*64+32,"O";
 360 Y=USR(0):L=L+1:IFPEEK(15364+X)<>32,2000
 370 PRINT@X,B$;:GOTO 300
```

Megszakításvezérelt manómozgatás C64-re

Manókat használó játékprogramjainknál — és különösen az ablaktechnikát alkalmazó felhasználói programoknál — sokszor szükség lehet arra, hogy a képernyőn egy manót, esetleg egy kurzort a program futásától függetlenül botkormánnyal mozgathassunk. Ezt a feladatot oldja meg az alábbi program úgy, hogy lehetővé teszi a nullás manók mozgatását az egész képernyőn. Négy bájt azt is meghatározza, hogy a monitor mely részén mozoghassunk a manóval. A bájtcímek: $CF24, $CF46, $CF31, $CF68, $033C. Az első kettő a bal felső, a második kettő a jobb alsó sarkát adja meg a manó mozgására rendelkezésre álló ablaknak. A programban ezek úgy vannak beállítva, hogy a manó bal felső sarka mindig a látható képernyőterületen marad. A programcímen állandóan ØØ-t tart, és ennek mértékét csak azokban a pillanatokban változtatja meg FF-re, amikor lenyomjuk a botkormány tűzgombját.

OLÁH GERGELY

```
CFØØ 78        SEI
CFØ1 AD 14 Ø3  LDA $Ø314
CFØ4 8D 8A CF  STA $CF8A
CFØ7 A9 19     LDA #$19
CFØ9 8D 14 Ø3  STA $Ø314
CFØC AD 15 Ø3  LDA $Ø315
CFØF 8D 8B CF  STA $CF8B
CF12 A9 CF     LDA #$CF
CF14 8D 15 Ø3  STA $Ø315
CF17 58        CLI
CF18 6Ø        RTS
CF19 EA        NOP
CF1A AD ØØ DC  LDA $DCØØ
CF1D 4A        LSR
CF1E BØ ØA     BCS $CF2A
CF2Ø AE Ø1 DØ  LDX $DØØ1
CF23 EØ 32     CPX #$32
CF25 FØ Ø3     BEQ $CF2A
CF27 CE Ø1 DØ  DEC $DØØ1
CF2A 4A        LSR
CF2B BØ ØA     BCS $CF37
CF2D AE Ø1 DØ  LDX $DØØ1
CF3Ø EØ F9     CPX #$F9
CF32 FØ Ø3     BEQ $CF37
CF34 EE Ø1 DØ  INC $DØØ1
CF37 4A        LSR
CF38 BØ 1F     BCS $CF59
CF3A AA        TAX
CF3B AD 1Ø DØ  LDA $DØ1Ø
CF3E 6A        ROR
CF3F 8A        TXA
CF4Ø EØ Ø7     BCS $CF49
CF42 AE ØØ DØ  LDX $DØØØ
CF45 EØ 18     CPX #$18
CF47 FØ 1Ø     BEQ $CF59
CF49 CE ØØ DØ  DEC $DØØØ
CF4C DØ ØB     BNE $CF59
CF4E AA        TAX
CF4F AD 1Ø DØ  LDA $DØ1Ø
CF52 4A        LSR
CF53 18        CLC
CF54 2A        ROL
CF55 8D 1Ø DØ  STA $DØ1Ø
CF58 8A        TXA
CF59 4A        LSR
CF5A BØ 1F     BCS $CF7B
CF5C AA        TAX
CF5D AD 1Ø DØ  LDA $DØ1Ø
CF6Ø 6A        ROR
CF61 8A        TXA
CF62 9Ø Ø7     BCC $CF6B
CF64 AE ØØ DØ  LDX $DØØØ
CF67 EØ 57     CPX #$57
CF69 FØ 1Ø     BEQ $CF7B
CF6B EE ØØ DØ  INC $DØØØ
CF6E DØ ØB     BNE $CF7B
```

```
1000 PRINT"* * * * * * * * *"
1001 PRINT"*       ,       *"
1002 PRINT"*   B A Z I S   *"
1003 PRINT"*    - - - - -  *"
1004 PRINT"* * * * * * * * *"
1090 PRINT:PRINT"HA MEHET NYOMJON A-T !"
1100 IF INKEY$<>"A",1100
1110 CLS:GOTO 200
1200 X=X-64:FORI=1TO3:IFPEEK(F+X+I)<>32,2000 ELSE NEXT
1210 PRINT@X+65,"    ";:RETURN
1300 X=X+64:FORI=1TO3:IFPEEK(F+X+I)<>32,2000 ELSE NEXT
1310 PRINT@X-63,"    ";:RETURN
1500 M=1:N=X+15362
1510 POKEN,32:N=N+64:E=E+.5:E1=INT(E*E-3)
1520 OUT 31,1:OUT30,2:OUT31,0:OUT30,E1
1530 IFPEEK(N)=88,POKEN,32:P=P+1:GOTO 1570
1540 IFPEEK(N)=191,1570
1550 IF PEEK(N)=32,POKEN,73:N=N-1:RETURN
1570 M=0:E=7:OUT31,1:OUT30,0:OUT31,0:OUT30,0:RETURN
2000 IFX<960,PRINT@X+65,"    ";
2005 IFX>64,PRINT@X-63,"    ";
2007 OUT31,7:OUT 30,255
2010 PRINT@X,CHR$(177);CHR$(138);CHR$(160);CHR$(163);
2020 FORJ=1 TO 2000:NEXT:CLS
2030 PRINT"ON HIBAZOTT !";L+(P*30);" PONTOT ERT EL."
2040 PRINT:PRINT
2050 INPUT"OHAJT MEG JATSZANI";A$
2060 IF A$<>"I",END ELSE RESTORE:L=0:GOTO 30
3000 DATA"0021013C11003C01E303EDB0213F3C11400060F36201910FB"
3005 DATA"211F3C36832AF87F3AFB7F47ED5FE603281E3D281C3D2809"
3010 DATA"78FE012809190518978FE0A28F7B7ED52047832FB7F"
3015 DATA"22F87FE536BF1910FBE1ED5FE603C0ED523658C9"
3050 DATA 0,0,0,0,0,0,3,28,15,15,0,0,0,0
```

ZX–Spectrum gyors feltöltése

A BASIC nyelv végrehajtási ideje kissé lassú. Az alábbi program a gyors képernyőfeltöltést segíti elő.

```
10 FOR i=16384 TO 16384+6912
20 POKE i,255
30 NEXT i
```

Mint látható, ez a program a képernyő- és színmemóriát tölti fel 255-ös kóddal (BIN 11111111).

Minden bizonnyal 58 másodperc alatt végezhető el a művelet. Ezt követően közöljük a gépi kódú programot a beolvasóval. A jártasabbak disassemblálhatják a kódot. (A tömörség kedvéért az assembler programot nem közöljük.)

```
10 FOR i=65000 TO 65015
20 READ a
30 POKE i,a
40 NEXT i
50 DATA 33,255,63,1,1,27,22,BIN
   11111110,35,11,120,177,200,114,24,
   248
60 RANDOMIZE USR 65000
```

Ez utóbbinál a feltöltési idő 1 másodperc. Tehát már a sebesség miatt is érdemes a gépi kódú programozással próbálkozni.

KEMÉNY ANDRÁS

```
CF70 AA        TAX
CF71 AD 10 D0  LDA $D010
CF74 4A        LSR
CF75 38        SEC
CF76 2A        ROL
CF77 8D 10 D0  STA $D010
CF7A 8A        TXA
CF7B 4A        LSR
CF7C A2 00     LDX #$00
CF7E 8E 3C 03  STX $033C
CF81 B0 05     BCS $CF88
CF83 A2 FF     LDX #$FF
CF85 8E 3C 03  STX $033C
CF88 EA        NOP
CF89 4C 31 EA  JMP $EA31
.:CF00 78 AD 14 03 8D 8A CF A9
.:CF08 19 8D 14 03 AD 15 03 8D
.:CF10 8B CF A9 CF 8D 15 03 58
.:CF18 60 EA AD 00 DC 4A B0 0A
.:CF20 AE 01 D0 E0 32 F0 03 CE
.:CF28 01 D0 4A B0 0A AE 01 D0
.:CF30 E0 F9 F0 03 EE 01 D0 4A
.:CF38 B0 1F AA AD 10 D0 6A 8A
.:CF40 B0 07 AE 00 D0 E0 18 F0
.:CF48 10 CE 00 D0 D0 0B AA AD
.:CF50 10 D0 4A 18 2A 8D 10 D0
.:CF58 8A 4A B0 1F AA AD 10 D0
.:CF60 6A 8A 90 07 AE 00 D0 E0
.:CF68 57 F0 10 EE 00 D0 D0 0B
.:CF70 AA AD 10 D0 4A 38 2A 8D
.:CF78 10 D0 8A 4A A2 00 8E 3C
.:CF80 03 B0 05 A2 FF 8E 3C 03
.:CF88 EA 4C 31 EA FF FF FF FF
```

Hangzó billentyűk Plus/4-re és C16-ra

A program minden billentyű lenyomásakor hangot ad, közben programot írhatunk. Ha folyamatosan nyomjuk a billentyűt, ugyancsak folyamatos hang szól. Viszont, ha valaki ezt nem akarja, akkor írjon a $ 0135 bájtra FF-t.

Indítás BASIC-ből SYS 290-nel.

MIHÁLYI GÁBOR

```
. 0123 A9 2E     LDA #$2E
. 0125 8D 14 03  STA $0314
. 0128 A9 01     LDA #$01
. 012A 8D 15 03  STA $0315
. 012D 60        RTS
. 012E A6 C6     LDX $C6
. 0130 E0 40     CPX #$40
. 0132 F0 0F     BEQ $0143
. 0134 A9 FE     LDA #$FE
. 0136 8D FC 04  STA $04FC
. 0139 A9 FF     LDA #$FF
. 013B 0D FE 04  STA $04FE
. 013E A9 18     LDA #$18
. 0140 8D 11 FF  STA $FF11
. 0143 4C 0E CE  JMP $CE0E
```

Pályázat számítástechnikai klubok részére

A középfokú oktatási intézmények számítástechnikai klubjainak munkáját technikai feltételeik javításával is szeretnénk támogatni, ezért részükre mikroszámítógépeket, illetve kiegészítő berendezéseket adunk használatba, meghatározott időre:

minden tanévben

szeptember 1-től június 10-ig.

E támogatás pályázat útján nyerhető el, melyet minden évben **április 31-ig** kell beküldeni a következő címre:

KISZ KB KSZTT

1388 Budapest

Pf.: 72.

A pályázatok tartalmazzák a következőket:

- a klubot működtető szerv, iskola nevét és címét,
- a klubfoglalkozások, illetve a nyitva tartás helyét, idejét
- a klubvezető(k) nevét,
- a klub célkitűzéseit, esetleg programját,
- a klubtagság, illetve a klub látogatásának feltételeit,
- a rendelkezésre álló gépek számát, típusát, és hogy milyen kiegészítő berendezések segítenék legjobban a klub munkáját.

KISZ KB KSZTT

C64

Óra a keretben

Az alábbi program egy mindig leolvasható, hat számjegyű órát varázsol a C64 alsó keretének jobb szélére. Az óra folyamatosan jár, látható az órák, percek, másodpercek értéke. Ez olyan BASIC-programokkal egybeépítve lehet hasznos, ahol szükség van a pontos idő állandó kijelzésére, miközben a teljes képernyő használható marad. Hátránya, hogy a lemezegységet ki kell iktatni, mint mindig, amikor sprite-ok vannak a képernyőn bekapcsolva. A hat számjegyet ugyanis két spritc, a 7. és 8. jeleniti meg. A kijelzett idő a CIA 1. chip belső órája, tehát indításkor ezt kell beállitani. A program eleje mutatja, hogyan lehet ezt BASIC-programmal megtenni.

A program észreveszi a hibásan beirt adatsorokat, és kiirja a hiba helyének sorszámhatárait. Az idő kijelzése közben a BASIC-utasítások futási ideje csak mintegy 3 százalékkal lesz hosszabb.

GYURKOVITS GERGELY

```
1 REM
10 REM )DERKO(  C=64 ORA A KERETBEN
20 REM
30 REM  IRTA: GYURKOVITS GERGELY  1987.
40 REM
50 C=52735:FOR J=1TO4:S=0:FOR I=1TO80
60 C=C+1:READA$:H=ASC(A$)-48:H=H+(H)9)*7
70 L=ASC(RIGHT$(A$,1))-48:L=L+(L)9)*7
80 H=H*16+L:S=S+H:POKEC,H:NEXT I:READ L
90 IF S=L THEN NEXT J:C=56329:GOTO 110
100 GOTO 900
110 POKE C+6,PEEK(C+6) AND 127
120 PRINT"KEREM A PONTOS IDOT ";
130 PRINT"(ORA,PERC,SEC) FORMABAN !"
140 INPUT O,P,S
150 IF O)23 OR P)59 OR S)59 THEN 120
160 IF O(0  OR P(0  OR S(0  THEN 120
170 IF O)11 THEN O=O OR 128
180 POKE C+2,O+INT(O/10)*6
190 POKE C+1,P+INT(P/10)*6
200 POKE C+0,S+INT(S/10)*6
210 PRINT"NYOMJ (RETURN)-T, HA A BEIRT"
220 PRINT"IDO EGYEZIK A VALOSSAL !"
230 POKE 53029,0:REM MINDENT KIIRJON
240 INPUTI:SYS 52736:INPUTI:REM START
250 POKE 53029,ASC("GYG")
260 REM NEM IR MINDENT KI.
270 END
500 DATA 78,A9,3E,8D,14,03,A9,CE
510 DATA 8D,15,03,A9,7F,8D,0D,DC
520 DATA AD,0E,DC,09,80,8D,0E,DC
530 DATA A2,0E,8E,FE,07,E8,8E,FF
540 DATA 07,A2,14,8D,2A,CF,9D,09
550 DATA D0,CA,D0,F7,8E,2D,D0,8E
560 DATA 2E,D0,8E,0B,DC,8E,FF,3F
570 DATA CA,8E,12,D0,58,60,AD,19
580 DATA D0,4A,90,5C,AD,12,D0,C9
590 DATA F6,90,48,AD,11,D0,A8,29
595 DATA 10270
600 DATA F7,8D,11,D0,AD,21,D0,85
610 DATA 02,CE,66,CE,98,A0,06,88
620 DATA D0,FD,AE,20,D0,A0,0C,88
630 DATA 8E,21,D0,30,09,AC,12,D0
640 DATA C0,FC,B0,11,90,F7,A0,19
650 DATA 8C,66,CE,F0,08,48,20,A3
660 DATA CE,AD,08,DC,68,8D,11,D0
670 DATA A9,32,8D,12,D0,EE,19,D0
680 DATA 4C,7E,EA,A5,02,8D,21,D0
690 DATA A9,F9,8D,12,D0,EE,19,D0
695 DATA 10702
700 DATA 4C,31,EA,AD,0F,DC,29,7F
710 DATA 8D,0F,DC,AD,09,DC,A2,CF
720 DATA 20,EA,CE,A2,D0,20,F1,CE
730 DATA 20,22,CF,AD,0A,DC,A2,90
740 DATA 20,EA,CE,A2,CE,20,F1,CE
750 DATA 20,22,CF,AD,0B,DC,10,0A
760 DATA 29,7F,C9,12,F0,0A,F8,69
770 DATA 12,D8,C9,12,D0,02,A9,00
780 DATA A2,8E,20,EA,CE,A2,8F,4C
790 DATA F1,CE,8D,20,CF,4A,4A,4A
795 DATA 10435
800 DATA 4A,8E,12,CF,29,0F,0A,0A
810 DATA 0A,18,69,B0,8D,0F,CF,A9
820 DATA D0,69,01,8D,10,CF,A6,01
830 DATA A9,33,85,01,A0,07,B9,A8
840 DATA D1,99,C0,03,CE,12 "F,CE
850 DATA 12,CF,88,10,F1,86,01,A9
860 DATA 55,60,F0,06,A9,47,F0,02
870 DATA 68,68,60,77,FF,1B,03,30
880 DATA 03,C0,1B,C0,46,3D,C0,C8
890 DATA 00,15,79,F1,00,00,00,00
895 DATA 8133
900 PRINT"HIBA A"400+100*J"-"495+100*J
910 PRINT"SORSZAMU DATA BLOKKBAN !":END
920 REM 39540 A TFLJES OSSZEG
```

RAINBOW
Számítástechnikai és Szolgáltató Kisszövetkezet

D–Subminiatur csatlakozók, kábelek, analóg-digitál átalakítók, egyéb számítástechnikai berendezések és kellékek nagy választékával várja Önöket

a MAGÉV–RAINBOW Elektronikai Szaküzlet

Budapest VI., Rudas László u. 33.
Telefon: 122-392. Telex: 22-6323

RAINBOW Számítástechnikai és Szolgáltató Kisszövetkezet

Központ: Budapest, Szilágyi Erzsébet fasor 1. 1026
Levélcím: 1378 Budapest 64., Postafiók 31. Telefon: 352-558

BASIC és gépi kód

Legutóbb a beszúrásokat és azok egyik legelterjedtebb változatát, a BASIC és KERNAL vektorainak átírását ismertettem. Most megnézzük, hogyan tudunk a meglévőn kívül további USR függvényeket definiálni és használni. Előtte — a témával összefüggésben — a tokenekről ejtek néhány szót.

A tokenek

A BASIC interpreter az utasítások begépelése és a RETURN billentyű lenyomása után az utasításokat tokenizálja. Ez azt jelenti, hogy az utasításban előforduló kulcsszavakat (utasítás- és függvényneveket, műveleti jeleket) azok egybájtos kódjával, tokenjével helyettesíti. Ezáltal a tárolt program mérete és végrehajtási ideje rövidebb lesz, de az utasítások feldolgozása is egyszerűbbé válik.

```
033c a9 51      lda #$51
033e a0 03      ldy #$03
0340 8d 08 03   sta $0308
0343 8c 09 03   sty $0309
0346 a9 9e      lda #$9e
0348 a0 03      ldy #$03
034a 8d 0a 03   sta $030a
034d 8c 0b 03   sty $030b
0350 60         rts
```

1. lista

A tokenek belső kódja $80-nál nem kisebb érték, így egy tokenizált BASIC-sor elemzésekor előjelvizsgálattal felismerhetjük, hogy tokenről van-e szó. Természetesen egyéb körülményeket is figyelembe kell venni, melyekre most nem térek ki.

A mostani gépi kódú rutinokban három token valamelyikének meglétét vizsgáljuk: a DEF ($96), az USR ($B7) és az egyenlőségjel ($B2) tokenjéét.

A programról

A listákon látható program C64-re készült, és a bevezetőben vázolt feladat megoldásának gondolatmenetét mutatja be.

Az *1. listán* a BASIC vektorokat átíró rutin található. A *2. listán* van a DEF USR utasítást feldolgozó rutin. Az IBM PC BASIC-jétől eltérően definiálni tudjuk az eredeti, sorszám nélküli USR függvényt is, tehát összesen 11 különböző felhasználói függvény alkalmazására van lehetőségünk. A *3. listán* az USR függvényeket kiértékelő rutin látható.

A DEF USR-t feldolgozó rutin működésének vázlata a következő. A rutin megvizsgálja, hogy az utasítás a DEF tokenjével kezdődik-e. Ha nem, a feldolgozás az eredeti interpreter rutinban folytatódik. DEF esetén megvizsgálja, hogy az USR tokenje következik-e. Ha nem, akkor az interpreternek a DEF FN utasítást feldolgozó rutinjába ugrik. Ellenkező esetben számjegy és egyenlőségjel vagy azonnali egyenlőségjel következhet, minden másra SYNTAX ERROR hibaüzenetet kapunk. Ha nincs hiba, akkor az egyenlőségjel jobb oldalán levő aritmetikai kifejezés kiértékelése és a címtáblázatba, illetve — sorszám nélküli USR esetén — az USR vektorba történő lerakása, majd az interpreter feldolgozó ciklusába való visszaugrás következik.

2. lista

```
0351 20 73 00   jsr $0073
0354 c9 96      cmp #$96
0356 f0 06      beq $035e
0358 20 79 00   jsr $0079
035b 4c e7 a7   jmp $a7e7
035e 20 73 00   jsr $0073
0361 c9 b7      cmp #$b7
0363 f0 03      beq $0368
0365 4c b3 b3   jmp $b3b3
0368 85 02      sta $02
036a 20 73 00   jsr $0073
036d f0 68      beq $03d7
036f b0 08      bcs $0379
0371 29 0f      and #$0f
0373 0a         asl
0374 85 02      sta $02
0376 20 73 00   jsr $0073
0379 c9 b2      cmp #$b2
037b d0 5a      bne $03d7
037d 20 73 00   jsr $0073
0380 20 8a ad   jsr $ad8a
0383 20 f7 b7   jsr $b7f7
0386 a6 02      ldx $02
0388 30 0a      bmi $0394
038a 9d db 03   sta $03db,x
038d 98         tya
038e 9d da 03   sta $03da,x
0391 4c ea a7   jmp $a7ea
0394 8d 12 03   sta $0312
0397 98         tya
0398 8d 11 03   sta $0311
039b 4c ea a7   jmp $a7ea
```

```
039e a9 00      lda #$00
03a0 85 0d      sta $0d
03a2 20 73 00   jsr $0073
03a5 c9 b7      cmp #$b7
03a7 d0 0f      bne $03b8
03a9 20 73 00   jsr $0073
03ac f0 29      beq $03d7
03ae 90 0e      bcc $03be
03b0 a5 7a      lda $7a
03b2 d0 02      bne $03b6
03b4 c6 7b      dec $7b
03b6 c6 7a      dec $7a
03b8 20 79 00   jsr $0079
03bb 4c 8d ae   jmp $ae8d
03be 29 0f      and #$0f
03c0 0a         asl
03c1 48         pha
03c2 20 73 00   jsr $0073
03c5 20 f1 ae   jsr $aef1
03c8 68         pla
03c9 a8         tay
03ca b9 da 03   lda $03da,y
03cd 85 55      sta $55
03cf b9 db 03   lda $03db,y
03d2 85 56      sta $56
03d4 4c e0 af   jmp $afe0
03d7 4c 08 af   jmp $af08
```

3. lista

A címtáblázat egy húszbájtos terület, amely a tíz számozott USR függvény belépési pontjának címét tartalmazza, természetesen csak definiálás után. Indulásnál — az USR vektorhoz hasonlóan — az ILLEGAL QUANTITY ERROR üzenetet kiíró ROM rutin címével ($B248) kell feltölteni. A példaprogramban a táblázat közvetlenül a program után van elhelyezve, és nincs feltöltve az említett rutin címével. Ha valaki a mostani lista alapján vállalkozik a program kipróbálására, vegye ezt figyelembe.

A 3. listán látható kiértékelő rutin működési elve hasonló az előbbiéhez. Először megvizsgálja, hogy a feldolgozás alatti aritmetikai vagy karakterlánc-kifejezés következő eleme az USR tokenje-e. Ha nem, akkor a feldolgozás az eredeti ROM rutinon folytatódik. Ugyanez történik — némi módosítással —, ha az USR tokent nem számjegy követi. Sorszámos USR függvény esetén a feldolgozás a függvény kiértékelésével folytatódik.

A fenti leírás csak vázlatosan ismerteti a rutin működését. A részletes magyarázattal és a BASIC betöltőprogramok listájával legközelebb jelentkezem.

BARNA LÁSZLÓ

Z80 programok haladóknak Spectrumra és Primóra

9. Manószerkesztő program II.

```
;-------------------------;
;                         ;
;      SPRITEDEF 32*32    ;
;       UHI-software      ;
;        1987.07.11       ;
;                         ;
;-------------------------;
        ORG   $E800
        JP    SPRDEF
;-------------------------;
SCR     EQU   $4000
SCBUF   EQU   $D000
BUFEND  EQU   $E7FF
SCRLEN  EQU   $1800
SCREND  EQU   $57FF
AADD    EQU   $484A
AHADD   EQU   $8A
BADD    EQU   $4010
A00     EQU   $4050
B00     EQU   $33B0
D00     EQU   $0F00
INK     EQU   16
PAP     EQU   17
AT      EQU   22
CR      EQU   13
EOL     EQU   $80
PIXADD  EQU   $22AC
EMIT    EQU   $0010
UDG     EQU   $5C7B
BSP     EQU   $8F00
SIZ     EQU   32

TXT     DEFS  2
FIFO    EQU   $-1
        DEFS  50*3
FIFOE   EQU   $-1
READER  DEFS  2
WRITER  DEFS  2
SPRADD  DEFS  2
FILEN   DEFS  2
SP1     DEFS  1
SPE     DEFS  1
SPOS    DEFS  2
POS     DEFS  2
CDS     DEFS  2
INBUF   DEFS  6
SBUFF   DEFS  128
PATTS   DEFS  64
;-------------------------;
CLEAR   CALL  SAVESP
CLEAR1  CALL  MSG
        DEFB  PAP,7,INK,0
        DEFB  19,1,EOL
        LD    BC,$0A0A
CLEAR2  CALL  ATBC
        CALL  MSG
        DEFM  "
        DEFB  EOL
        INC   B
        LD    A,B
        CP    14
        JR    NZ,CLEAR2
        LD    BC,$0010
CLEAR3  CALL  ATBC
        LD    A,6
        CALL  EMIT
        INC   B
        LD    A,B
        CP    16
        JR    NZ,CLEAR3
        CALL  MSG
        DEFB  19,0,EOL
        RET
;-------------------------;
PLOT    LD    BC,(POS)
        CALL  BCOORD
        LD    D,4
        LD    E,4
PLOT1   LD    A,C
        SUB   4
        LD    C,A
        DEC   B
        DEC   D
        JR    NZ,PLOT1
        LD    BC,(POS)
        CALL  ACOORD
        CALL  PLOTBC
        RET
;-------------------------;
BREAK   EI
        LD    (IY+83),$38
        LD    (IY+84),$00
        LD    (IY+14),$38
        RST   8
        DEFB  $FF
;-------------------------;
LEFT    LD    A,(POS)
        DEC   A
        AND   $1F
        LD    (POS),A
        RET
;-------------------------;
RIGHT   LD    A,(POS)
        INC   A
        AND   $1F
        LD    (POS),A
        RET
;-------------------------;
DOWN    LD    A,(POS+1)
        DEC   A
        AND   $1F
        LD    (POS+1),A
        RET
;-------------------------;
UP      LD    A,(POS+1)
        INC   A
        AND   $1F
        LD    (POS+1),A
        RET
;-------------------------;
SIGN    LD    HL,(POS)
        LD    (SPOS),HL
        RET
;-------------------------;
NEW     LD    C,3
        CALL  INP
        DEFM  "To wich "
        DEFM  "sprite? "
        DEFB  EOL
        CALL  SPRNO
        RET   C
        CALL  NEWS1
        CALL  LISTSP
        RET
;-------------------------;
BACK    LD    A,(SP1)
        OR    A
        RET   Z
        DEC   A
        JR    SPRN1
;-------------------------;
FORW    LD    A,(SP1)
        INC   A
        LD    B,A
        LD    A,(SPE)
        SUB   7
        CP    B
        RET   C
        LD    A,B
        JR    SPRN1
;-------------------------;
ROT     LD    HL,AADD
        LD    (ADDR1),HL
        INC   HL
        INC   HL
        INC   HL
        LD    (ADDR4),HL
        LD    B,SIZ-1
ROT1    CALL  HLDOWN
        DJNZ  ROT1
        LD    (ADDR3),HL
        DEC   HL
        DEC   HL
        DEC   HL
        LD    (ADDR2),HL
        LD    A,$80
        LD    (MASK1),A
        LD    (MASK2),A
        RLCA
        LD    (MASK3),A
        LD    (MASK4),A
        LD    C,16
ROT2    LD    B,16
ROT3    PUSH  BC
        LD    HL,ADDR1
        CALL  EXBIT
        AND   $1F
        LD    (POS+1),A
        RET
;-------------------------;
SIGN    LD    HL,(POS)
        LD    (SPOS),HL
        RET
;-------------------------;
NEW     LD    C,3
        CALL  INP
        DEFM  "To wich "
        DEFM  "sprite? "
        DEFB  EOL
        CALL  SPRNO
        RET   C
        CALL  NEWS1
        CALL  LISTSP
        RET
;-------------------------;
BACK    LD    A,(SP1)
        OR    A
        RET   Z
        DEC   A
        JR    SPRN1
;-------------------------;
FORW    LD    A,(SP1)
        INC   A
        LD    B,A
        LD    A,(SPE)
        SUB   7
        CP    B
        RET   C
        LD    A,B
        JR    SPRN1
;-------------------------;
ROT     LD    HL,AADD
        LD    (ADDR1),HL
        INC   HL
        INC   HL
        INC   HL
        LD    (ADDR4),HL
        LD    B,SIZ-1
ROT1    CALL  HLDOWN
        DJNZ  ROT1
        LD    (ADDR3),HL
        DEC   HL
        DEC   HL
        DEC   HL
        LD    (ADDR2),HL
        LD    A,$80
        LD    (MASK1),A
        LD    (MASK2),A
        RLCA
        LD    (MASK3),A
        LD    (MASK4),A
        LD    C,16
ROT2    LD    B,16
ROT3    PUSH  BC
        LD    HL,ADDR1
        CALL  EXBIT
```

Ez alkalommal befejezem a manószerkesztő ismertetését és egyben a sorozatot is. A *listán* közlöm a program még hiányzó részét és a teljes változótáblát. Ügyelni kell arra, hogy semmi ne szerepeljen kétszer. A teljes forráslista tehát a következő részekből áll:

9. rész: parancsrutinok
8. rész: programtörzs, néhány alaprutin
6-7. rész: I/O és egyéb alaprutinok
5. rész: mintás feltöltés (FILL)
4. rész: zárt alakzat festése az 5. részbeli változtatásokkal
1. rész: gyors LDIR, ahogyan a 4. részben szerepel is.

A kiszolgáló (parancs-) rutinok működése

A kiszolgáló rutinok közül a legbonyolultabb a ROT, amely 90 fokkal elforgatja a manót. Működésére nem térek ki, ráérő olvasók megfejthetik. Érdemes azonban megfigyelni, milyen gyorsan működik. Azért ilyen hosszú és bonyolult.

Tanácsok a fordításhoz, főleg Spectrumosoknak

A forrásszöveg meglehetősen hosszú, de ha elég alacsony címre töltjük, befér a

```
        CALL EXB1T
        CALL EXB1T
        CALL EXB1T
        LD   HL,ADDR1
        CALL EXB1T
        CALL NEXT
        POP  BC
        DJNZ ROT3
        CALL CORR
        DEC  C
        JR   NZ,ROT2
        CALL ATOB
        RET
EXB1T   LD   E,(HL)
        INC  HL
        LD   D,(HL)
        INC  HL
        LD   B,(HL)
        INC  HL
        LD   A,(DE)
        PUSH AF
        AND  B
        EX   AF,AF'
        POP  AF
        JR   Z,EXB0
        OR   B
        JR   EXBE
EXB0    OR   B
        XOR  B
EXBE    LD   (DE),A
        EX   AF,AF'
        RET
NEXT    LD   A,(MASK1)
        RRCA
        LD   (MASK1),A
        JR   NC,NEXT2
        LD   HL,(ADDR1)
        INC  HL
        LD   (ADDR1),HL
NEXT2   LD   HL,(ADDR2)
        CALL HLUP
        LD   (ADDR2),HL
NEXT3   LD   A,(MASK3)
        RLCA
        LD   (MASK3),A
        JR   NC,NEXT4
        LD   HL,(ADDR3)
        DEC  HL
        LD   (ADDR3),HL
NEXT4   LD   HL,(ADDR4)
        CALL HLDOWN
        LD   (ADDR4),HL
        RET
CORR    LD   HL,(ADDR1)
        DEC  HL
        DEC  HL
        CALL HLDOWN
        LD   (ADDR1),HL
        LD   HL,(ADDR2)
        LD   B,16
```

```
CORR21  CALL HLDOWN
        DJNZ CORR21
        LD   (ADDR2),HL
        LD   A,(MASK2)
        RRCA
        LD   (MASK2),A
        JR   NC,CORR3
        INC  HL
        LD   (ADDR2),HL
CORR3   LD   HL,(ADDR3)
        INC  HL
        INC  HL
        CALL HLUP
        LD   (ADDR3),HL
        LD   HL,(ADDR4)
        LD   B,SIZ/2
CORR41  CALL HLUP
        DJNZ CORR41
        LD   (ADDR4),HL
        LD   A,(MASK4)
        RLCA
        LD   (MASK4),A
        RET  NC
        DEC  HL
        LD   (ADDR4),HL
        RET
ADDR1   DEFS 2
MASK1   DEFS 1
ADDR2   DEFS 2
MASK2   DEFS 1
ADDR3   DEFS 2
MASK3   DEFS 1
ADDR4   DEFS 2
MASK4   DEFS 1
;-----------------------;
TURN    LD   HL,AADD
        LD   B,SIZ
TURN1   LD   D,H
        LD   E,L
        INC  DE
        INC  DE
        INC  DE
        CALL TURNB
        DEC  DE
        INC  HL
        CALL TURNB
        DEC  HL
        CALL HLDOWN
        DJNZ TURN1
        JP   ATOB
TURNB   PUSH BC
        LD   B,8
        LD   C,(HL)
        LD   A,(DE)
TURN2   RR   (HL)
        RLA
        RR   C
        DJNZ TURN2
        LD   (HL),C
        LD   (DE),A
```

```
        POP  BC
        RET
;-----------------------;
INV     LD   HL,AADD
        LD   C,SIZ
        LD   B,SIZ/8
INV1    LD   A,(HL)
INV2    CPL
        LD   (HL),A
        INC  HL
        DJNZ INV2
        DEC  HL
        DEC  HL
        DEC  HL
        DEC  HL
        CALL HLDOWN
        DEC  C
        JR   NZ,INV1
        JP   ATOB
;-----------------------;
RTS     LD   HL,(SPOS)
        LD   (POS),HL
        RET
;-----------------------;
DEL     LD   HL,AADD
        LD   DE,SBUFF
        LD   B,$20
DEL1    CALL EXX1
        CALL EXX1
        CALL EXX1
        CALL EXX1
        DEC  HL
        DEC  HL
        DEC  HL
        DEC  HL
        CALL HLDOWN
        DJNZ DEL1
        JP   ATOB
EXX1    LD   A,(DE)
        LD   C,(HL)
        EX   DE,HL
        LD   (DE),A
        LD   (HL),C
        EX   DE,HL
        INC  DE
        INC  HL
        RET
;-----------------------;
OLD     CALL SAVESP
        LD   C,3
        CALL INP
        DEFM "Wich old "
        DEFM "sprite? "
        DEFB EOL
        CALL SPRNO
        RET  C
        LD   BC,$4F50
        EX   DE,HL
        CALL SPRITE
        JP   ATOB
```

GENS3 mellé, és le tudja fordítani. Természetesen az egészet megjegyzések nélkül kell begépelni. Az assembler célszerű betöltési címe 23800. Fordításkor a „table space?" kérdésre 2000-et kell válaszolni.

A másik megoldás, hogy kazettáról fordítunk, de ez elég lassú, úgyhogy nem javaslom.

A program használata

A manószerkesztő üzenetei angol nyelvűek, egyrészt mivel a programot elsősorban programozók használják, akik ismerik a „compu-angolt", másrészt így nem kellett ékezetes betűkkel szerencsétlenkedni.

```
361 ;-------------------------;
362 FILLH   CALL LISTPT
363         CALL SAVESP
364         CALL GRIDB
365         CALL OPENIN
366         CALL MSG
367         DEFM "Choose a "
368         DEFM "pattern.."
369         DEFB EOL
370         CALL GETNO
371         CALL PATTNO
372         RET  C
373         LD   BC,(POS)
374         CALL ACOORD
375         EX   DE,HL
376         CALL FILL
377         CALL ATOB
378         CALL LISTSP
379         RET
380 GRIDB   LD   HL,AADD
381         CALL HLUP
382         DEC  HL
383         LD   C,SIZ+2
384 GRIDB1  PUSH HL
385         LD   B,SIZ/8+2
386 GRIDB2  LD   (HL),$FF
387         INC  HL
388         DJNZ GRIDB2
389         POP  HL
390         CALL HLDOWN
391         DEC  C
392         JR   NZ,GRIDB1
393         CALL DEL
394         CALL SAVESP
395         RET
396 ;-------------------------;
397 PATT    CALL LISTPT
398         CALL OPENIN
399         CALL MSG
400         DEFM "To wich "
401         DEFM "pattern? "
402         DEFB EOL
403         CALL GETNO
404         CALL PATTNO
405         RET  C
406         LD   DE,AADD
407         EX   DE,HL
408         LD   B,8
409 PATT1   LD   A,(HL)
410         LD   (DE),A
411         INC  DE
412         CALL HLDOWN
413         DJNZ PATT1
414         CALL LISTPT
415         CALL KEYIN
416         CALL LISTSP
417         CALL EXEC
418         RET
419 ;-------------------------;
420 HELP    CALL MSG
```

```
421         DEFB 19,1,PAP,7,EOL
422         LD   HL,HELPM
423         LD   B,0
424 HELP1   LD   C,16
425         CALL ATBC
426 HELP2   LD   A,(HL)
427         CALL EMIT
428         INC  HL
429         DEC  C
430         JR   NZ,HELP2
431         INC  B
432         LD   A,B
433         CP   16
434         JR   NZ,HELP1
435         CALL PAUSE0
436         CALL ATOB
437         RET
438 HELPM   DEFM "5&8 cursor move"
439         DEFM "0 plot    S sign"
440         DEFM "V clear   B back"
441         DEFM "R rotate  N forw"
442         DEFM "P pattern       "
443         DEFM "I inverse L line"
444         DEFM "D delete  F fill"
445         DEFM "T return to sign"
446         DEFM "D elete last    "
447         DEFM "  command       "
448         DEFM "M irror         "
449         DEFM "SPACE   to basic"
450         DEFM "ENTER  if ready!"
451         DEFM "                "
452         DEFM "To load or save "
453         DEFM " retun to basic!"
454 ;-------------------------;
```

A program elindításakor megkérdezi a manóalakfájl kezdőcímét. Erre azért van szükség, mert a programot elvben bárhova fordíthatjuk, és hozzá képest bárhol elhelyezhetjük a memóriában a manóalakfájlt. Nem ellenőrzi, úgyhogy ügyeljünk az elhelyezésre. Ezután megkérdezi, hány manót szeretnénk szerkeszteni. Az így megadott manóalakfájl címét és hosszát ki is írja a képernyőre.

A kész fájl kimentése úgy történhet, hogy a SPACE billentyűvel kilépünk BASIC-be és BASIC utasítással kimentjük. A cím és a hossz segítségként a képernyőn látható.

Az egyes szerkesztőbillentyűk az előző rész 1. táblázatában láthatók. Ehhez azonban néhány kiegészítést kell fűzni. A kurzormozgató billentyűk elhelyezkedése azért éppen olyan amilyen, mert Spectrumon így a Cursor Joystick használható erre a célra. Akinek ilyen nincs, vagy Primo-tulajdonos, az átírhatja a TABLE1-ben és a HELPM szövegben a szükséges karaktereket úgy, hogy kényelmesebb legyen.

Ha kész vagyunk egy manóval, nyomjuk meg az ENTER-t. Ekkor megkérdezi, hogy a manóalakfájl hányadik elemére tegye az új manót, majd odamásolja. Szükség esetén új automata listát is kapunk (lásd fent: SPRNO rutin), hogy mindjárt megjelenjen az új manó a szemünk előtt is az „O" ablakban.

A D (törlés) billentyű második lenyomása a törlés előtti állapotot hozza vissza.

Mintás feltöltéshez új mintázatot a következőképpen lehet csinálni. Mivel a minták 8×8 bitesek (csak az ismétlődés miatt látszanak nagyobbnak a mintalistában), a szerkesztőmező bal felső sarkába rajzoljunk meg egy 8×8-as mintát. Hogy a mező többi részén mi van, az mindegy. Ha kész, nyomjuk meg a P (pattern = mintázat) gombot. Erre megkérdezi, hogy melyikre tegye (0 . . . 7). Közben kiteszi az „O" ablakba a régieket, hogy lássuk, melyiket töröljük.

A forrásszöveg nem tartalmaz kész mintákat, úgyhogy azokat mind meg kell szerkeszteni. Célszerű „teli" mintát is csinálni, gyakran kell. Ha ezután felvesszük a programot, a minták „vele mennek".

Ha a 2. részben bemutatott manókezelőhöz manót szerkesztünk, akkor mindegyikhez külön takarási maszkra van szükség. Ezeket úgy kell megrajzolni, mint közönséges manókat. Erre a következő módszert javaslom. Tegyük fel, hogy manónknak zárt körvonala van. Először a manó látható alakját rajzoljuk meg, és rögzítsük a manóalakfájlban, majd a rajta kívül maradt felületet kenjük be teljesen a megfelelő „teli" mintával. Ha a manó hozzáért a 32×32-es keret széléhez, akkor ezt a műveletet a kívül maradt felület minden elemére végezzük el. A kész maszkot a látható alak elé kell tenni. A 256-tal osztható kezdőcímre nem kell ügyelni, csak akkor, amikor használjuk.

Primósoknak

Jó hír, hogy semmit nem kell változtatni, mivel elég magas szintű rutinoknál tartunk: a gépfüggő alaprutinokon már túl vagyunk.

Lehet változtatni a PICT képernyőrajzoló rutinon: ki lehet irtani belőle a színvezérlő karaktereket. Sajnos a színek hiánya miatt az ablakok nem fognak olyan szépen elkülönülni, mint a Spectrumon.

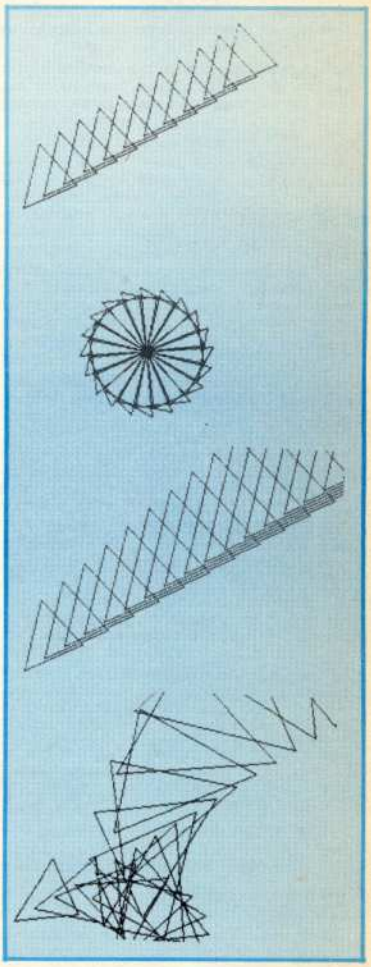
UHERKOVICH PÉTER

Közületek, figyelem! Mikroszámítógépet akarnak vásárolni? Tájékozódjanak a naprakész piaci helyzetről! Díjtalan ismertető! MESZ Számítástechnika 1368 Budapest, Pf. 193.

Számítógépes grafika Pascalban

SÍK TRANSZFORMÁCIÓK

Az előzőekben már láthattunk példát arra, hogyan lehet a képernyő egyes „ablakait", vagyis karakterhelyekkel vagy képpontokkal határolt téglalap alakú területeit a képernyő egy másik részére átmásolni, 90°-kal elforgatni, tükrözni stb. Ezeket a műveleteket összefogaló néven transzformációknak hívják.



Sokkal általánosabban használható és főleg sokkal gyorsabban működő eljárásokat fejleszthetünk ki, ha a transzformációkat nem a képernyőn megjelenített raszterpontokon végezzük el közvetlenül, hanem előbb elkészítjük az ábrázolandó kép geometriai modelljét, majd ennek fontosabb pontjain hajtjuk végre a transzformációt, és ezt rajzoltatjuk ki a számítógéppel.

A geometriai modell legyen egyszerű és mégis eléggé rugalmas ahhoz, hogy tetszőleges vonalas rajzolatokat tudjunk vele leírni. Ehhez pontokat (a mintaprogramban legfeljebb 50 pontot), valamint ezeket a pontokat a megadás sorrendjében összekötő egyenes szakaszok nyílt vagy záródó sorozatát használunk.

A pontok koordinátáinak tárolására a COORD azonosítójú rekordtípust deklaráltuk a főprogramban, melynek mezőlistája egy X és egy Y integert tartalmaz. A pontok koordinátáinak eredeti értékeit és transzformáltjait is a POINTS azonosítójú rekordtömbben fogjuk tárolni.

Ezek után térjünk rá a transzformációkra. Ha egy geometriai alakzatot önmagával párhuzamosan el akarunk tolni valamelyik koordinátatengely (például x) irányában, mondjuk Tx értékkel, nem kell mást tennünk, csak minden pontjának x koordinátájához Tx-et hozzáadnunk. Természetesen egyidejűleg y irányban is végezhetünk eltolást vagy transzlációt; ilyenkor a tényleges elmozdulás valamilyen ferde irányban fog bekövetkezni. Ha a transzformált koordinátákat vesszővel jelöljük, a transzformáció során általában ezt a két műveletet kell végrehajtani minden pontra:

$x' = x + Tx$

$y' = y + Ty$

A másik elemi transzformációs művelet a geometriai alakzat elforgatása a koordinátarendszer kezdőpontja körül. E sorozat egyik korábbi cikkében megmutattuk, hogy ha α szöggel akarjuk a pontot elforgatni, akkor az új koordináták így számíthatók:

$x' = x.\cos\alpha - y.\sin\alpha$

$y' = x.\sin\alpha + y.\cos\alpha$

A forgatást idegen szóval rotációnak is nevezik.

Végül a harmadik elemi transzformáció a léptékezés, mellyel nagyítani, kicsinyíteni lehet. A léptékezés transzformációs egyenletei:

$x' = x.Sx$

$y' = y.Sy$

Ezzel igen sok érdekes transzformáció hajtható végre. Ha például $Sx = Sy = 2$, akkor az eredeti ábránkat kétszeresére nagyítjuk. Ha $Sx = Sy = 0{,}5$, a képet felére kicsinyítjük. $Sx = 1$, $Sy = 2$ csak függőleges irányban nyújt, míg $Sx = 1$, $Sy = -1$ az y tengelyre tükröz stb.

Ha az olvasó gondolatban kipróbálja, milyen műveletek valósíthatók meg azáltal, hogy az elemi transzformációk paramétereinek adunk más-más értéket, a lehetséges változatok egész sorát kapja. Van mégis néhány feladat, amely egy transzformációval nem valósítható meg. Ha például a képer-

```
 10 {$L-}
 20 {1:TRANSZ filenev alatt kimentett INCLUDE rutinok}
 30
 40 PROCEDURE LineSet(N,Address:INTEGER);
 50 {Egyenes szakaszokbol allo nyilt lanc}
 60 VAR I,CIM,X0,Y0:INTEGER;
 70     XY          :COORD;
 80 BEGIN
 90   XY:=PEEK(Address,COORD);
100   WITH XY DO
110   BEGIN
120     X0:=X;
130     Y0:=Y
140   END;
150   CIM:=Address+4;
160   FOR I:=1 TO (N-1) DO
170   BEGIN
180     XY:=PEEK(CIM,COORD);
190     WITH XY DO
200     BEGIN
210       PixelLine(X0,Y0,X,Y);
220       X0:=X;
230       Y0:=Y
```

```
240     END;
250     CIM:=CIM+4
260   END
270 END; {LineSet}
280
290 PROCEDURE Polygon(N,Address:INTEGER);
300 {Sokszog rajzolasa}
310 VAR x0,y0:INTEGER;
320     XY   :COORD;
330 BEGIN
340   LineSet(N,Address);
350   XY:=PEEK(Address,COORD);
360   WITH XY DO
370   BEGIN
380     x0:=X;
390     y0:=Y
400   END;
410   XY:=PEEK(Address+4*(N-1),COORD);
420   WITH XY DO
430     PixelLine(x0,y0,X,Y)
440 END; {Polygon}
450
460 PROCEDURE Zero(VAR M:Matrix);
```

```
470 (3*3 matrix nullazasa)
480 VAR i,j:INTEGER;
490 BEGIN
500   FOR i:=1 TO 3 DO
510   FOR j:=1 TO 3 DO
520     M[i,j]:=0
530 END; (Zero)
540
550 PROCEDURE Translation(VAR T:Matrix;Tx,Ty:REAL);
560 (Eltolasi matrix kepzese)
570 VAR i:INTEGER;
580 BEGIN
590   Zero(T);
600   FOR i:=1 TO 3 DO
610     T[i,i]:=1;
620   T[3,1]:=Tx;
630   T[3,2]:=Ty
640 END; (Translation)
650
660 PROCEDURE Rotation(VAR T:Matrix;Alfa:REAL);
670 (Forgato matrix kepzese)
680 VAR sina,cosa:REAL;
690 BEGIN
700   Zero(T);
710   sina:=SIN(Alfa);
720   cosa:=COS(Alfa);
730   T[1,1]:=cosa;
740   T[2,2]:=cosa;
750   T[3,3]:=1;
760   T[2,1]:=sina;
770   T[1,2]:=-sina
780 END; (Rotation)
790
800 PROCEDURE Scaling(VAR T:Matrix;Sx,Sy:REAL);
810 (Nagyito-kicsinyito matrix kepzese)
820 BEGIN
830   Zero(T);
840   T[1,1]:=Sx;
850   T[2,2]:=Sy;
860   T[3,3]:=1
870 END; (Scaling)
880
890 PROCEDURE MatrixMult(VAR M1,M2,M3:Matrix);
900 (3*3 matrixok szorzasa)
910 VAR i,j,k:INTEGER;
920     s     :REAL;
930 BEGIN
940   FOR i:=1 TO 3 DO
950     FOR j:=1 TO 3 DO
960     BEGIN
970       s:=0;
980       FOR k:=1 TO 3 DO
990         s:=s+M1[i,k]*M2[k,j];
1000      M3[i,j]:=s
1010    END;
1020    M1:=M3
1030 END; (MatrixMult)
1040
1050 PROCEDURE VectorMult(VAR V1,V2:Vector;VAR M:Matrix);
1060 (3*3 matrix es sorvektor szorzasa)
1070 VAR i,j:INTEGER;
1080     s  :REAL;
1090 BEGIN
1100   FOR i:=1 TO 3 DO
1110   BEGIN
1120     s:=0;
1130     FOR j:=1 TO 3 DO
1140       s:=s+V1[j]*M[j,i];
1150     V2[i]:=s
1160   END;
1170   V1:=V2
1180 END; (VectorMult)
1190
1200 PROCEDURE Trans(n,Cimtol,Cimre:INTEGER);
1210 (n pont transzformacioja)
1220 VAR i,c1,c2:INTEGER;
1230     Pontxy :COORD;
1240 BEGIN
1250   c1:=Cimtol;
1260   c2:=Cimre;
1270   FOR i:=1 TO n DO
1280   BEGIN
1290     Pontxy:=PEEK(c1,COORD);
1300     WITH Pontxy DO
1310     BEGIN
1320       v1[1]:=X;
1330       v1[2]:=Y;
1340       v1[3]:=1;
1350       VectorMult(v1,v2,m1);
1360       X:=ROUND(v1[1]);
1370       Y:=ROUND(v1[2]);
1380     END;
1390     POKE(c2,Pontxy);
1400     c1:=c1+4;
1410     c2:=c2+4
1420   END
1430 END; (Trans)
```

1. lista

2. lista

```
10 {$L-}
20 PROGRAM SikTranszformacio;
30 {1987. februar 22.}
40 TYPE COORD =RECORD
50                X,Y:INTEGER
60              END;
70      Matrix=ARRAY [1..3,1..3] OF REAL;
80      Vector=ARRAY [1..3] OF REAL;
90 VAR POINTS :ARRAY [1..100] OF COORD;
100     N,I,J,n     :INTEGER;
110     v1,v2  :Vector;
120     m1,
130     m2,m3  :Matrix;
140
150 {$L+}
160 {$F 1:LINE  }
170
180 {$L+}
190 {$F 1:TRANSZ}
200
210 BEGIN {Foprogram}
220   PAGE;
230   WRITE('A pontok szama:');
240   READLN;READ(n);
250   WRITELN;
260   FOR I:=1 TO n DO
270     WITH POINTS[I] DO
280     BEGIN
290       WRITELN;
300       WRITE('x[',I,CHR(8),']=');
310       READLN;READ(X);
320       WRITE('y[',I,CHR(8),']=');
330       READLN;READ(Y)
340     END;
350   PAGE;
360   LineSet(n,ADDR(POINTS));
370   REPEAT UNTIL INCH<>CHR(0);
380   PAGE;
390   Poligon(n,ADDR(POINTS));
400   FOR I:=1 TO 10 DO
410   BEGIN
420     Translation(m1,I*15,I*10);
430     Trans(n,ADDR(POINTS),ADDR(POINTS[n+1]));
440     Poligon(n,ADDR(POINTS[n+1]));
450   END;
460   REPEAT UNTIL INCH<>CHR(0);
470   PAGE;
480   FOR I:=1 TO 20 DO
490   BEGIN
500     Rotation(m1,I*3.1415926/10);
510     Translation(m2,100,70);
520     MatrixMult(m1,m2,m3);
530     Trans(n,ADDR(POINTS),ADDR(POINTS[n+1]));
540     Poligon(n,ADDR(POINTS[n+1]));
550   END;
560   REPEAT UNTIL INCH<>CHR(0);
570   PAGE;
580   FOR I:=1 TO 20 DO
590   BEGIN
600     Scaling(m1,1+I*0.1,1+I*0.1);
610     Translation(m2,I*15,I*10);
620     MatrixMult(m1,m2,m3);
630     Trans(n,ADDR(POINTS),ADDR(POINTS[n+1]));
640     Poligon(n,ADDR(POINTS[n+1]));
650   END;
660   REPEAT UNTIL INCH<>CHR(0);
670   PAGE;
680   FOR I:=1 TO 20 DO
690   BEGIN
700     Scaling(m1,1+I*0.1,1+I*0.1);
710     Rotation(m2,I*3.1415926/10);
720     MatrixMult(m1,m2,m3);
730     Translation(m2,I*15,I*10);
740     MatrixMult(m1,m2,m3);
750     Trans(n,ADDR(POINTS),ADDR(POINTS[n+1]));
760     Poligon(n,ADDR(POINTS[n+1]));
770   END;
780   REPEAT UNTIL INCH<>CHR(0)
790 END.
```

nyő közepére felrajzolunk egy kört és ebből nagyítással szeretnénk egy másik, az elsővel koncentrikus kört kapni, akkor egy egyszerű léptékezés nem vezetne célhoz, mert a kör középpontja nem esik a koordinátarendszer kezdőpontjába, így a léptékezés a középpont origótól való távolságát is kinagyítaná: a két kör nem maradna koncentrikus. Ha az eredeti feladatot akarjuk megvalósítani, a következő lépéseket kell végrehajtani:

— El kell tolni a kör középpontját az origóba (vagyis az x=0, y=0 pontba, azaz a képernyő bal alsó sarkába).

— Az új koordinátákat másodszor is transzformálni kell: most jöhetne a megfelelő nagyítás.

— Végül egy harmadik transzformációval (transzlációval) vissza kell tolnunk a kör középpontját az eredeti helyére.

A transzformációk algoritmusát mátrixok segítségével számítógépekre alkalmasabb módon is megfogalmazhatjuk. Ha az olvasó esetleg idegenkedik ettől a középiskolában nem tanított matematikai eszköztől, átugorhatja a következőket; elég, ha a később ismertetendő Pascal-eljárások alkalmazásának szabályait tanulmányozza át a program ismertetésekor.

Nos, a transzformációkhoz egy pont koordinátáit háromelemű sorvektorba gyűjtjük. Ezzel az összes transzformáció az alábbi márixvektor-szorzással írható le:

$$[x'\ y'\ 1] = [x\ y\ 1] \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ e & f & 1 \end{bmatrix}$$

ahol a jobb oldali mátrix az ún. transzformációs mátrix, melynek itt betűkkel jelölt elemei attól függnek, milyen transzformációról van szó.

A mátrix alakja eltolás esetén:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ Tx & Ty & 1 \end{bmatrix}$$

forgatásnál:

$$\begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

léptékezésnél:

$$\begin{bmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Ha két mátrixot összeszorzunk, az eredmény ugyancsak egy mátrix, melynek annyi sora lesz, amennyi a bal oldali mátrixnak és annyi oszlopa, amennyi a jobb oldali mátrixnak van. A szorzatmátrix elemeit a következő egyszerű séma szerint lehet kiszámítani. Ha a jobb oldali mátrix alá írjuk a kiszámítandó szorzatvektort, akkor egyes elemeit úgy kell meghatározni, hogy a keresett elemmel egy sorban levő jobb oldali mátrix elemeit jobbról balra haladva megszorozzuk a bal oldali mátrix azonos oszlopába eső elemekkel felülről lefelé haladva, és a szorzatokat összegezzük.

$$\begin{bmatrix} 1 & 3 & 2 \\ 0 & 2 & -1 \\ 2 & -2 & 2 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 3 & - \\ 2 & 1 & - \end{bmatrix} \begin{bmatrix} 4 & 10 & 3 \\ 0 & 6 & 1 \end{bmatrix}$$

Például a megjelölt elem értéke így jön ki:
2.1+3.0+1.2=4

Ez az algoritmus könnyen programozható. Ha négyzetes mátrixot négyzetes mátrixszal szorzunk össze, az eredmény is négyzetes mátrix lesz, ha pedig vektort szorzunk vektorral, a szorzat is vektor lesz. Ha több transzformációt akarunk egymás után végrehajtani, akkor vagy annyiszor szorozzuk meg a pont koordinátavektorait a transzformációs mátrixszal, ahány transzformációt kell végrehajtani, vagy először összeszorozzuk egymással a transzformációs mátrixokat, majd a mátrixvektor-szorzásokat minden vektorra csak egyszer alkalmazzuk. Belátható, hogy ha sok egymást követő transzformációt kell sok pontra alkalmazni, akkor ez utóbbi megoldással sok felesleges műveletet takaríthatunk meg.

Az *1. lista* annak az include-fájlnak a listája, amelyben a transzformációhoz szükséges eljárásokat deklaráltuk. A *2. lista* demonstrációs programot tartalmaz ezeknek az eszközöknek a használatára.

A LineSet és a Poligon eljárás egyenes szakaszok nyílt láncát, illetve sokszöget rajzol. A Zero eljárás kinulláz egy 3×3-as négyzetes mátrixot. (A Sinclair BASIC-kel ellentétben ugyanis a Pascalban a tömb deklarációjakor csak helyfoglalás történik, a tömböt a program nem tölti fel nullákkal!)

A Translation, Rotation és Scaling eljárások a kinullázott transzformációs mátrix értékes elemeit töltik fel eltolás, forgatás és léptékezés esetére. A Matrixmult eljárás két 3×3-as négyzetes mátrixot szoroz össze úgy, ahogy azt az előbb bemutattuk, a VectorMult pedig négyzetes mátrix és vektor szorzatát álltja elő.

A Trans eljárás végrehajtja a tulajdonképpeni koordinátatranszformációt az egyes pontokra, és az eredményeket elhelyezi a memóriába.

A demonstrációs program beolvassa a felhasználó által megadott n darab pont koordinátáit, majd az ezekből szerkesztett sokszöget felrajzolja a képernyőre. Egymás után bemutatja az eltolás, a forgatás, az eltolás és nagyítás, valamint az eltolás, nagyítás, forgatás kombinációját. Példaképpen bemutatjuk, hogy egy egyszerű háromszög esetén a program milyen képeket állít elő. A főprogram — úgy gondoljuk — nem igényel magyarázatot.

DR. KABOLDY PÉTER

ADOK-VESZEK-CSERÉLEK

Ebben a rovatban rövid, szöveges, a mikroszámítógépekkel kapcsolatos hirdetéseket közlünk. A díjszabás: közleteknek gépelt soronként (60 karakter) 100,- Ft, magánszemélyeknek az első sor 50,- Ft, minden további sor 20,-Ft. Az NJSZT tagjainak az első három sor ingyenes. Hirdetéseiket a szerkesztőség címére várjuk.

ADOK

Seikosha GP 50 S printer Sinclair csatlakozóval eladó. Kristóf Attila, Zalaegerszeg, Dózsa György u. 2. 8900.

Commodore VC20 felhasználói és játékprogramok eladása. Kérjen tájékoztatót! Juhász György, Salgótarján, 3100. Pf.: 157.

Primo monitor minden memóriakiépítésű gépre! Gépi kódú program kimentés-betöltés, nyomkövetés, disassembler és sok más hasznos funkció. Ára leírással együtt, kazettán 170,- Ft, utánvéttel. A64-es gépre programokat cserélek. Ajánlatom: Assembler, Pascal, Lisp, FORTH, C rendszerek. Paller Gábor, Budapest, Varga Gyula András park 8/a. 1149.

Alig használt, jó állapotban lévő Videoton TVC családi okok miatt sürgősen eladó. Major István, Pécel, Széchenyi u. 9. 2119.

C64, VC20 stb. mikroszámítógépekhez mozgó alkatrész nélküli, "nem elromló", szenzoros botkormányt készítek. Egy év garancia! Ára 1000 forint/db. Megrendelhető postán: Varga Csaba, Szeged, Kereszttöltés 28/A. fsz. 2. 6723.

Commodore 64-hez GYORS-HÁTTÉRTÁR cartridge. Kapacitása 2-31 kbájt. A GYORS-HÁTTÉRTÁR-ba maximum 7 db, célszerűen gyakran használt program vihető be. A gép bekapcsolása után menüvel jelentkezik be, és gombnyomásra a kiválasztott program azonnal fut. Javasolt programcsomagok: Turbo tape, Assembler, Monitor, Supergrafik, Help plus, Turbo másoló. File másoló 1699 Ft és Turbo tape 699 Ft.
Hozott programok elhelyezése a GYORS-HÁTTÉRTÁR-ban. Trompler László, Budapest, Attila u. 22. 1201. Tel.: 287-493 este.

VESZEK

Primo típusú számítógépet vásárolok. Tartozékok, printer, szakkönyvek, programkazetták is érdekelnek. Árajánlatokat válaszborítékkal a következő címre kérek: Varsányi Gábor, Nagyatád, Aradi u. IX/C. IV.3. 7500.

Vennék üzemképtelen VC/VIC-20, C16, C116, C64, C Plus/4 számítógépeket és tartozékaikat, valamint VC-1541-es-lemezegységet. Darabok is érdekelnek. Maróti Gyula, Bordány, Zákányszéki út 22. 6795.

CSERÉLEK

Enterprise gépemet Sinclair QL-re vagy Schneider CPC 6128-ra cserélném. Varga Zoltán, Székesfehérvár, Lövölde u. 9/C. 8000.

ZX-Spectrum 48 k-ra készült színvonalas programokat cserélnék (Space Harrier, Feud, Xeno stb.). A válaszokat programlistával várom. Selyem Attila, Lengyeltóti, Fonyódi u. 51. 8693.

48 k-s ZX-Spectrumom Interface II-vel, Kempston interfésszel, 200 db játék- és felhasználói programmal, német és magyar nyelvű kézikönyvvel együtt elcserélném egy Commodore Plus/4-re, magnóval vagy C16-ra belső bővítéssel. Kiss György, Budapest, Juhász Gyula u. 34. IV.15. 1039. 17 óra után

Bináris fák II.

A fán értelmezhető műveletek

A bináris fa tárgyalásánál használt fogalmak

A bináris fának az I. rész végén (lásd előző számunkat) megadott definíciójában bevezetett jelölések mellett tételezzük fel, hogy *T, T1, T2* nem üres fák. Legyenek a felsorolt fák gyökerei rendre: *t, t1,t2*. Ekkor azt mondjuk, hogy:

— *T1* és *T2* fa a *T* fa részfái. A sorrend általában nem közömbös, legyen *T1* a *T* fa bal részfája, *T2* a jobb részfája.

— *t1* bal fia, *t2* jobb fia *T*-nek.

— *t* apja *t1*-nek és *t2*-nek.

— *t* az első szinten van *T*-ben, *t1* és *t2* a második szinten. Ha valamely csúcspont az *i*-edik szinten van *T*-ben, akkor fia vagy fiai az *i+1*-edik szinten.

— *Levele* a fának az a csúcspont, aminek nincs fia.

— Egy csúcspont *úttávolsága* azonos a csúcs *szintszámával*. Ebből következően azonos a gyökértől a csúcspontig útbaejtett csúcsok számával, a gyökért és az adott csúcsot is beszámítva. A úttávolság *gráffal* való ábrázolásánál az út éleinek száma +1 azonos a csúcs szintszámával.

— *T magassága* = csúcsai úttávolságának maximuma. Ez azonos a levelei úttávolságának vagy szintszámának maximumával.

Ha egy fát úgy építünk fel, hogy minden t_i csúcsra a bal részfán szereplő azonosítók kisebbek, a jobb részfa azonosítói pedig nagyobbak a t_i-hez tartozó azonosítónál, akkor a fát *keresőfának* nevezzük.

Bináris fa építése, keresés a fán

Keressük az *x*-szel jelölt azonosítót a fán!

Ha a fa üres, akkor *x* lesz a fa egyetlen eleme.

Ha nem üres, akkor a keresést a fa gyökeréből kiindulva kezdjük el. Ha az *y* csúcspontnál vagyunk, akkor a következő szabály szerint járunk el:

— Ha $x = y$, akkor az *y* azonosítót megtaláltuk a fán.

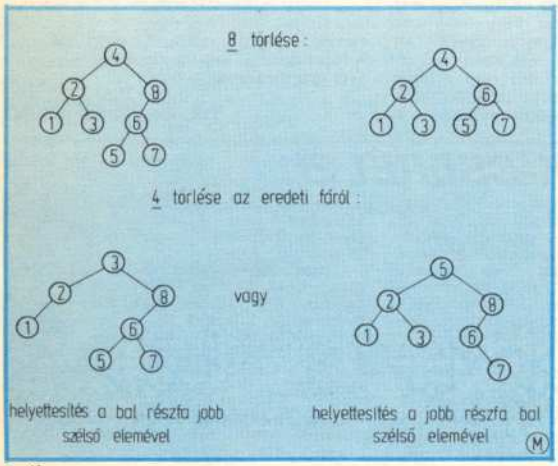
— Ha $x < y$, akkor *y* bal oldali fia lesz a következő útbaejtendő csúcspont, ha létezik. Ha nem létezik, akkor *x* nem szerepel a fán, és *y* bal oldali fiaként kell beilleszteni.

— Egyébként $x > y$ esetében *y* jobb oldali fia lesz az útbaejtendő csúcspont, ha létezik. Ha nem létezik, akkor *x* nem szerepel a fán, és *y* jobb oldali fiaként kell beilleszteni.

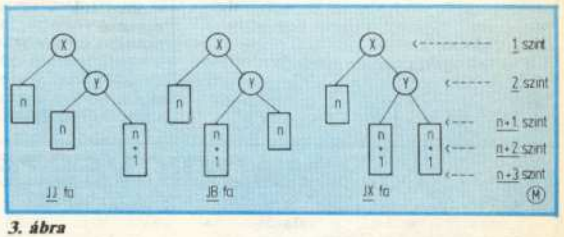
Fa bejárása

Előfordul, hogy valamilyen műveletet végre kell hajtani a fa minden csúcspontján. Ekkor egy algoritmus valamilyen sorrend szerint felkeresi a fa minden csúcsát, azaz bejárja a fát. Több bejárási mód létezik, számunkra az ún. *inorder (belső sorrendű) bejárás* a legfontosabb.

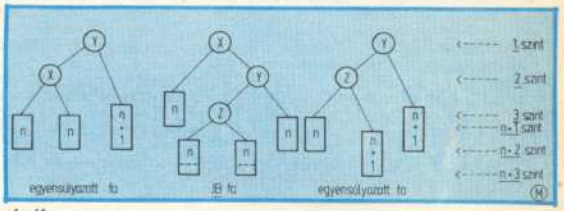
Maradva a bevezetett jelöléseknél, ez a rekurzióval a következő: *T* fa inorder bejárása = *T1* inorder bejárása, *t*, *T2* inorder be-



1. ábra



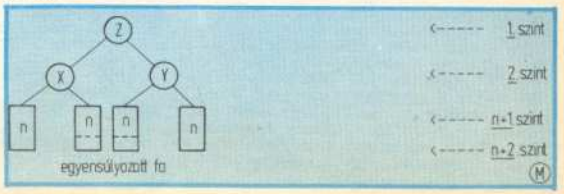
3. ábra

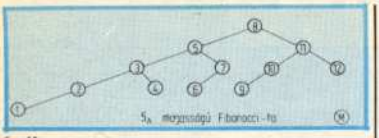


4. ábra

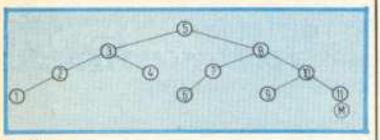
2. ábra

5. ábra





6. ábra



7. ábra

járása. Megjegyzésként annyit, hogy az inorder bejárás a *T* fa elemeinek növekvő felsorolását adja.

Törlés a fáról

Egy csúcspont kiiktatása valamivel bonyolultabb művelet, mint a beépítés. Ha a törlendő csúcspont vagy levél, vagy pedig egy utódja van, akkor az utód kapcsolódik a törlendő csúcs apjához. Ha két utód létezik, akkor két olyan elem is található, amelynek

— legfeljebb egy utóda van;

— rendezettség alapján beépülhet a törlendő elem helyébe.

Ilyen tulajdonságú a bal részfa jobb szélső eleme és a jobb részfa bal szélső eleme.

Például az I. részbeli 2. ábrán a barack helyébe léphet az ananász vagy a datolya. Egy másik példaként lásd itt az *1. ábrát*. Állapodjunk meg abban, hogy mindig a bal részfa jobb szélső csúcsával helyettesítünk.

Kiegyensúlyozott bináris fák

A fa felépítése erősen függ attól, hogy milyen sorrendben rakjuk fel rá az azonosítókat, a fa elemeit. Szerencsétlen esetben a fának bizonyos „ágai", útjai megnyúlhatnak. Ha egy eleve rendezett sorozatból építjük fel a fát, a fa egyetlen bal úttal vagy egyetlen jobb úttal rendelkezik — a rendezés irányától függően. A keresés során ilyen esetben a lánchoz képest is bonyolultabb a helyzet. El kell érni, hogy ilyen eset ne fordulhasson elő. Szerencsére van olyan faátalakítás, amelyet végrehajtva minden olyan beépítési vagy törlési szituáció után, amikor a fa valamely útja a többihez képest nagyon megnyúlik, a fát egy azonos rendezettségű, ún. *kiegyensúlyozott fává* vihetjük át.

Mit jelent az, hogy egy fa kiegyensúlyozott?

Több kiegyensúlyozási fogalom is létezik.

Tökéletesen kiegyensúlyozott a fa, ha minden csúcsnak a bal, illetve jobb részfájában található csúcsok száma legfeljebb *1*-gyel tér el.Ez igen szigorú követelmény, nehezen is teljesíthető. Célszerű ehelyett egy gyengébb kritériumot előírni, ami egyszerűbb átalakító eljárást eredményez, a keresés hatékonyságát pedig nem befolyásolja számottevően.

Ezek után az olyan bináris fát nevezzük *kiegyensúlyozottnak*, amelynek minden csúcsára igaz az, hogy a két részfa magasságának különbsége legfeljebb *1*. A feladat az, hogy az építés és a törlés műveleteknél fenntartsuk a kiegyensúlyozottságot. Kiegyensúlyozott fákon a keresést, beszúrást, csúcstörlést legfeljebb log,*n* nagyságrendű lépésszámmal lehet végrehajtani.

Egyensúlyozási műveletek

Beszúrás, törlés sőt egyensúlyozás során is keletkezhetnek olyan fák, amelyek egyik ága *2*-vel eltér a másiktól. A következő alapesetek fordulhatnak elő (lásd a 2—5. ábrákat).

Az egyensúlyát vesztett fa alakjai a *2. ábrán* láthatók: lehet jobbra billent (*J* típusú) fa, vagy balra billent (*B* típusú) fa. A szimmetria miatt elég csak a *J* fákat vizsgálnunk tüzetesebben. A *3. ábra* részletezi, hogy a jobb ág hogyan lehet *n+2* hosszú. Beszúrásnál csak *JJ* és *JB* fa jöhet létre. A *JB* fa is csak olyan lehet, amelyben a *Z* csúcshoz csatlakozó egyik részfa *1*-gyel magasabb, mint a másik. Törlésnél és az azt követő egyensúlyozásnál bármilyen fa szóba jöhet a felsorolt alakzatok közül. (*B* fáknál lehetnek *BB, BJ, BX* fák.) A *4. ábrán* láthatók a *JJ*, illetve a *JX* fa *JJ* forgatásai. A *JB* fa így nem nyerné vissza egyensúlyát, vizsgálatához további lebontás szükséges.

A fenti három csúcs kiemelése, majd az ún. *JB-forgatás* után az *5. ábrán* látható kiegyensúlyozott fát kapjuk meg.

A beszúrásnál egy leghosszabb út keletkezésével támad egyensúlyzavar, törlésnél ezzel szemben egy legrövidebb út keletkezésével.

Míg beszúrás után legfeljebb egy csúcsot kell egyensúlyozni, addig törlés után a törölt vagy az annak helyébe lépő csúcs keresőútjának minden csúcsa egyensúlyozásra szorulhat. Szerencsére ez ritkán fordul elő. Ilyen szempontból a legszerencsétlenebbek az ún. *Fibonacci-fák*. Ennek részletezésére most nem térek ki, hanem a *6. ábra* példáján bemutatok egy ilyet. A *12.* csúcs törlése két egyensúlyozást von maga után; törlés és kiegyensúlyozás után a fa a *7. ábrán* látható. (Folytatjuk.)

FLENDER ANDRÁS

COMMODORE 64

A képernyő balra forgatása

Érdekes és meglepő hatásokat érhetünk el, ha a képernyő tartalmát egy vagy több bájttal elforgatjuk. Az alábbi gépi kódú program (*1. lista*) ezt valósítja meg. Egy BASIC program képernyőre listázása után akár programból, akár közvetlenül a SYS52992 utasítás segítségével elindítva a gépi kódú programot, a programlista a képernyőn egy bájttal balra fordul. A programmal egyébként a teljes képernyőre kiterjedő képújságot is készíthetünk.

A program magyarázata

180—193 A szimbolikus gépi kódú fordítóprogram számára megadjuk a szükséges kezdőértékeket. A képernyő kezdőcíme a bekapcsoláskor érvényes helyen van: a képernyő az 1024 ($400) tárcímnél kezdődik, és 2023-nál ($7E7-nél) ér véget. A szint tároló RAM-terület is a szokásos helyen, 55296-nál kezdődik. Forgatáskor a szintárolóba 7-et, a sárga színkódját írjuk. AKT-tal az egyes képpontokat, SZM-mel a szintároló mutatóját jelöljük.

210—248 Kezdőértéket adunk a képernyő és a szintároló mutatóinak: AKT=$400, SZM=$D800.

260—280 Itt kezdődik az előkészület a forgatáshoz. Az egyes sorok legelső karakterét kiolvassuk és az X tárolóba tesszük.

290—320 A forgatási ciklus magja. Itt írjuk egy bájttal előbbre az egyes bájtok tartalmát.

322—323 A sárga szín beültetése a szintárolóba.

340—350 Vége van-e a sornak, a 40 karakternek (0—39)?

360—373 A kimentett karaktert az X tárolóból beírjuk a sor utolsó helyére, és a szintárolóban az utolsó karakternek is sárga színt adunk.

390—447 A képpont és a szintároló mutatójának értékét egy sorral, 40 karakterrel megnöveljük.

460—520 Vége van-e az utolsó sornak is a képernyőn? Ha még nincs, akkor a program visszaugrik a forgatóciklus elejére.

A *2. lista* a BASIC-betöltőt tartalmazza. Ez a program a memóriába más helyre is szabadon betölthető. Ilyenkor értelemszerűen a kezdőcímet és a végcímet is meg kell változtatni az 1000-es sorban, és megváltozik a program indítási (SYS) címe is: a mindenkori betöltési kezdőcímmel egyezik meg.

A gépi kódú program működését a *3. listán* látható BASIC program segítségével mutatjuk be. Ez a program az A$ szöveges változó tartalmát a képernyő második sorába és első oszlopába írja ki (lásd a 3. lista 2—3-as sorát), majd 1 bájttal az egész képernyőt balra elforgatja. Ezt a műveletet még 37-szer megismétli. A 8-as sor az így létrejött képernyőtartalmat háromszor körbeforgatja a teljes képernyőn. A 9-es sortól kezdve egy teljes képernyőre kiterjedő képújságot hoz létre. A 9-es sor átállítja a képernyőt 40-ről 38 karakteresre. A képújságszerűen megjelenítendő szöveg a DATA sorokban található. A 11—15-ös sorokban a program a képújság szövegének betűit egyenként beírja a képernyő első oszlopába, két helyre egymás alá (CHR$(17) a kurzort mozgatja lefelé, CHR$(19) a kurzort a képernyő kezdő pozíciójába viszi, CHR$(147) a képernyő törlését jelenti), és ezt követően a képernyőt egy bájttal balra forgatja. Így a letakart első oszlopba íródik az új információ, ahonnan a forgatás hatására átkerül az ugyancsak letakart utolsó oszlopba, majd a további forgatások hatására jelenik meg képújságszerűen a szöveg. Ha a programot RUN10-zel indítjuk el, akkor a képernyő

```
100: CF00           KEP     =   $400
105: CF00           KEPVEG  =   $7E7
106: CF00           SZINTAR =   $D000
107: CF00           SZIN    =   7
108: CF00           AKT     =   $FB
109: CF00           SZM     =   $FD
                    ;
199: CF00                   **  $CF00
                    ;
210: CF00 A9 00     INIT    LDA  #< KEP
220: CF02 A0 04             LDY  #> KEP
230: CF04 85 FB             STA  AKT
240: CF06 84 FC             STY  AKT+1
242: CF08 A9 00             LDA  #< SZINTAR
244: CF0A A0 D0             LDY  #> SZINTAR
246: CF0C 85 FD             STA  SZM
248: CF0E 84 FE             STY  SZM+1
                    ;
260: CF10 A0 00     FORGAS  LDY  #0
270: CF12 B1 FB             LDA  (AKT),Y
280: CF14 AA                TAX
290: CF15 C8        CIKLUS  INY
300: CF16 B1 FB             LDA  (AKT),Y
310: CF18 88                DEY
320: CF19 91 FB             STA  (AKT),Y
                    ;
322: CF1B A9 07             LDA  #SZIN
323: CF1D 91 FD             STA  (SZM),Y
330: CF1F C8                INY
340: CF20 C0 27             CPY  #$27
350: CF22 D0 F1             BNE  CIKLUS
360: CF24 8A                TXA
370: CF25 91 FB             STA  (AKT),Y
372: CF27 A9 07             LDA  #SZIN
373: CF29 91 FD             STA  (SZM),Y
                    ;
390: CF2B 18                CLC
400: CF2C A5 FB             LDA  AKT
410: CF2E 69 28             ADC  #$28
420: CF30 85 FB             STA  AKT
430: CF32 90 02             BCC  TOVI
440: CF34 E6 FC             INC  AKT+1
                    ;
442: CF36 18        TOVI    CLC
443: CF37 A5 FD             LDA  SZM
444: CF39 69 28             ADC  #$28
445: CF3B 85 FD             STA  SZM
446: CF3D 90 02             BCC  VEGVIZS
447: CF3F E6 FE             INC  SZM+1
                    ;
460: CF41 A5 FC     VEGVIZS LDA  AKT+1
470: CF43 C9 07             CMP  #> KEPVEG
480: CF45 D0 C9             BNE  FORGAS
490: CF47 A5 FB             LDA  AKT
500: CF49 C9 E7             CMP  #< KEPVEG
510: CF4B 90 C3             BCC  FORGAS
520: CF4D 60                RTS
```

1. lista

```
1000 FOR I= 52992  TO  53069
1010 READ X:POKE I,X:G=G+X:NEXT
1020 DATA 169,  0,160,  4,133,251,132,252
1030 DATA 169,  0,160,216,133,253,132,254
1040 DATA 160,  0,177,251,170,200,177,251
1050 DATA 136,145,251,169,  7,145,253,200
1060 DATA 192,  39,208,241,138,145,251,169
1070 DATA    7,145,253, 24,165,251,105, 40
1080 DATA 133,251,144,  2,230,252, 24,165
1090 DATA 253,105, 40,133,253,144,  2,230
1100 DATA 254,165,252,201,  7,208,201,165
1110 DATA 251,201,231,144,195, 96
1120 IF G<> 12415 THEN PRINT"HIBAS ADAT !":END
1130 PRINT"OK !!"
```

2. lista

```
0 PRINTCHR$(147)
1 A$="  SZABO PETER PAL    "
2 A=LEN(A$):PRINTA$CHR$(19)CHR$(17)CHR$(17)
3 FORI=1TOA:B$=MID$(A$,I,1):PRINTB$;:NEXT
4 SYS52992:REM KEPFORDATAB
5 FORJ=0TO49:NEXT
6 N=N+1:IFN=37THEN8
7 PRINTCHR$(19)CHR$(17)CHR$(17):GOTO3
8 FORI=0TO119:SYS52992:FORJ=0TO29:NEXT:NEXT
9 POKE53278,PEEK(53278)AND247
10 N=0:PRINTCHR$(147):GOSUB25
11 FORI=1TOA:PRINTCHR$(19)CHR$(17)CHR$(17)
12 B$=MID$(A$,I,1):PRINTB$CHR$(17)CHR$(17)
13 PRINTCHR$(17)CHR$(17)B$:SYS52992
14 FORJ=0TO99:NEXT:REM KESLELTETES
15 NEXT
16 N=N+1:IFN=2THEN18
17 PRINTCHR$(19)CHR$(17)CHR$(17):GOTO11
18 POKE53270,PEEK(53270)OR8:END
19 :
20 REM A KEPUJSAG ADATAI
21 :
22 DATA" A TELJES KEPERNYON LEHET EVVEL"
23 DATA" A PROGRAMMAL EGY KEPUJSAGOT"
24 DATA" MEGVALOGITANI!"
25 FORJ=1TO3:B$="":READB$:C$=C$+B$:NEXT
26 A#=C$:A=LEN(A$):RETURN
```

3. lista

átállítása elmarad. Így megfigyelhetjük, hogyan íródik be az első oszlopba az információ. A 25—26-os sorokban található alprogram olvassa ki a DATA sorokból az adatokat és helyezi el az A$ szöveges változóban.

Legközelebb a mozgatható objektumok, a sprite-ok forgatásáról lesz szó.

SZABÓ PÉTER PÁL

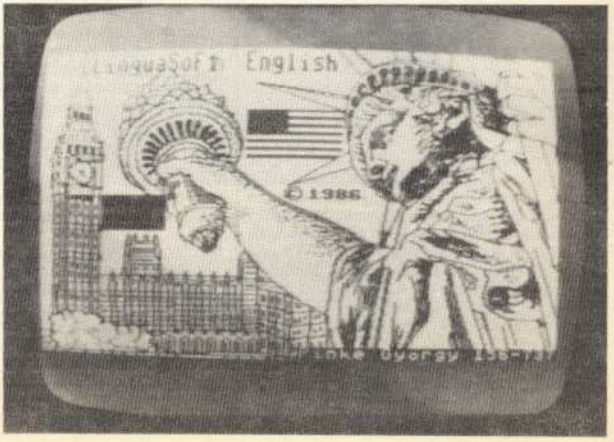
Mit tud a LINGUASOFT

A Software '88 kiállításon
újra találkozhattunk a µ '87-en
fődíjat nyert nyelvoktató programmal,
a LINGUASOFT-tal. A rendszer
alapötlete Gálai Antal bajai
mérnöktől származik. A nyelvi
anyagot Knáb Erzsébet,
Medgyes Péter és Albert Sándor,
a német, az angol és a francia nyelv
egyetemi oktatói állították össze,
nem kis munkával.
A LINGUASOFT az egyik
legelterjedtebb gépre, a C64-re
készült. Aki rászánja magát
a tanulásra, indulásnál négy
menü közül választhat.

Vizsgáztat, gyakoroltat

A vizsga üzemmódban haladhatunk a leggyorsabban, mert a rendszer ekkor csak osztályoz, vagyis méri a teljesítményt. Mondatonként és a teljes vizsga anyagára minősíti tudásunkat. Hasonlít a teszthez, azzal a különbséggel, hogy az eredményt azonnal megkapjuk.

A gyakorló üzemmódban már keményen meg kell küzdeni a továbblépésért, mert a helyes fordítás begépeléséig nem kapunk új mondatot. Igaz, kérhetünk segítséget, de ezért fizetni kell. Az ismeretlen szavakat adott tarifáért megkérdezhetjük a géptől: a képernyőn megjeleníti azok összes szinonimáját a gyakorolt nyelven. Ekkor már csak választani kell, de a begépeléstől nem menekülünk meg. Nagyon hosszan nem lehet gondolkozni, mert a türelmi idő leteltével segítségül megkapjuk a legjobb megoldást. Ez a szó a képernyőn inverz karakter formájában, szalagszerűen jelenik meg, figyelmeztetve a tanulót, hogy az nem a saját teljesítménye. Amikor közösen kiizzadtuk a



helyes fordítást, az eltűnik a képernyőről, és önállóan kezdhetjük elölről. Ha sikerül helyesen begépelnünk a fordítást, akkor játékosan megjelenik, hogy mennyi pénzt költöttünk, és jöhet az új mondat.

Gyakorlat teszi a mestert

Az edzés nagyon is érzékletes, mert a rossz vagy helytelenül begépelt szót a rendszer azonnal letörli a képernyőről, sőt még szirénáz is hozzá.

Sokunkkal előfordul, hogy a sikertelenségért a gépet okoljuk. Persze mindig neki van igaza. Számára a nagy- és kisbetű, a vessző és a pont felcserélése és elhagyása ugyanolyan hiba, mint a téves szó beírása. Ha kételkedünk is néha, végül a megoldást látva be kell ismernünk, hogy szigorú, de korrekt tanárral van dolgunk.

A fordítás beírásakor nem kell a képernyőt néznünk, mert hibáinkra azonnal figyelmeztet a már említett sziréna. Az idő múlását gongütések jelzik.

A LINGUASOFT szigorú, de türelmes is. A programba beépített függvények — az áramkörökben levő tanár — különbséget tesznek a gépelési bizonytalanságból és a nyelvi tudás hiányából eredő hiba között. A beépített súlyfüggvény a tanuló alkatához igazodik.

Dolgozatot is kell írni

A harmadik menü a dolgozatírás. Ez a verzió a negyedikkel együtt csoportok, tankörök számára készült. Negyven diák próbálkozását, tudását mérheti mágneses memóriájában. Ez az opció hasonlít a vizsga üzemmódhoz: a tanárnak elég időszakonként ellenőriznie a tanulók haladását ahhoz, hogy teljes képet kapjon arról, kivel mit kell gyakoroltatnia. Így több idő marad az élőbeszédre.

Nem bújhatunk el

Most említjük meg, hogy a LINGUASOFT-tal csak bejelentkezés után lehet tanulni. Személyi szám közlése kötelező! Utána már nem lehet letagadni a sikertelen próbálkozásokat, javításra azonban van mód. A rendszer a tanfolyam teljes ideje alatt személyenként tárolja az eredményeket.

A dolgozatjavítás a negyedik üzemmód. A program nemcsak fáradhatatlanul korrepetál, hanem mentesíti a tanárt a legriasztóbb tehertétel, a dolgozatjavítás alól. A tanár kényelmesen végignézheti, míg az árulkodó Júdás, a negyedik opció nagy sebességgel bemutatja, amit a tanulók a dolgozatírás során „elkövettek".

Hogyan működik?

Vajon honnan tudja mindezt a LINGUASOFT nyelvoktató program? Tudjuk, hogy adott mondat nagyon sokféleképpen fordítható le. Nem tévedhet a program egy jó, de különös fordítás elbírálásánál? És még számos hasonló kérdést tehetünk fel.

A LINGUASOFT alkotói természetesen nem állítják, hogy rendszerük tévedhetetlen. Mindenesetre a nyelvészek és programozók munkája több megméretésen jelesre vizsgázott.

A rendszer — többek között — szinonimaszótár és nyelvhelyességi alrendszerből épül fel. Hogy hogyan, az a fejlesztők titka!

A szó begépeléséig a LINGUASOFT csendben figyel. A szóköz lenyomása után a program megnézi a szótárában, hogy van-e ilyen szó. Ha megtalálja és nem észlel helyesírási hibát, akkor belép a nyelvtani, fordítási algoritmusok „épületébe", és eldönti, hogy a szó helyes sorrendben érkezett-e. Természetesen számtalan lehetséges esetet kell figyelembe vennie.

Egy kis statisztika

Jelenleg a rendszer kétszáz mondat fordításával gyakoroltat, illetve vizsgáztat. Az ezekhez szükséges összes szót, azok szinonimáját és nyelvhelyességi szabályait tartalmazza. A fordítási változatok száma néhánytól több ezerig terjed. A következő mondat például 122 880-féleképpen fordítható le: „Ugyanakkor az előadók semmit vagy csekély honoráriumot kaptak csak fellépéseikért."

A nyelvenkénti statisztikát táblázatba foglalva közöljük. Ez, ha nem is a teljesség igényével, de jól tükrözi a különböző idegen nyelvek „szabadságát", variációs lehetőségeit.

A program különbséget tesz fordítás és fordítás között. Az értékelésnél figyelembe veszi a megoldás élőnyelvre jellemző gyakoriságát, de nem bünteti a választékos kifejezéseket sem. Az elfogadható, de kissé erőltetett fordításoknál azonban már „fizethetünk" induló pénzecskénkből, amelyet a gyakorlás elején kapunk, az adott nyelv országának pénznemében.

A LINGUASOFT maximális figyelmet fordít a helyesírásra. Ismeri az ékezetes betűket: ezeket egyszerűen az 1-es melletti ← billentyű segítségével állíthatjuk elő. A „manipulációkhoz" nem kell különösebb mutatvány, könnyen megjegyezhetők. Ha azonban elfelejtjük vagy eltévesztjük valamelyiket, megszólal a hibát jelző sziréna.

Feleselő írógép

A LINGUASOFT-ot könnyű használni, nem kell hozzá előtanulmányokat folytatni. A menük megválasztásának módját és a funkcióbillentyűk hatását a képernyőről mindig leolvashatjuk.

A program erős középfokú nyelvtudásra épül, de használhatják kezdők is, mert bármelyik opcióban segítséget kérhetnek a szinonimaszótárból vagy a fordítás már beírt részének folytatásához.

A LINGUASOFT-tal betöltött Commodore-t feleselő írógépnek is hívják, mert a hibákra azonnal reagál és visszaszól. Igaz, a feleselő gyerekkel vagy feleséggel ellentétben ki lehet kapcsolni, de nem érdemes. Aki egyszer kipróbálta, nem tud leszokni róla.

P. GY.

A LINGUASOFT statisztikája

	angol	német	francia
Mondatok száma:	200	200	200
Magyar szavak száma:	1 637	1 748	1 707
Idegen szavak száma:	5 073	6 334	4 927
Összes fordítások száma:	261 644	28 626	33 146
Átlagos fordítások száma:	1 308	143	165

Konzervlevelek

Gálai Antalt
a µ '87 Országos Mikroszámítógépes
Találkozón ismertem meg.
Erről be is számoltam a Magazin
1987. júniusi számában.
A Software '88 kiállításon,
a LINGUASOFT bemutatóján újra
megkerestem őt.
A stand körül tolongó tömegben
nem volt könnyű a közelébe
férkőznöm, de végül sikerült
elrabolnom, hogy folytassuk
közel egy éve megkezdett
beszélgetésünket.

— Hogyan alakult a LINGUASOFT nyelvoktató rendszer sorsa a µ '87-en aratott siker, a fődíj elnyerése után?

— Már ott a kiállításon sok megrendelést vettünk fel. A sokszorosítás és másolás zökkenői miatt eleinte el is úsztunk egy kicsit a szállítással, de viszonylag hamar egyenesbe jöttünk. Most már rövid időn belül tudunk szállítani. Az érdeklődés nem lankad, folyamatosan kapunk megrendeléseket.

— Tervezték, hogy bővítik a tanulható nyelvek körét, elsősorban a nemzetiségi nyelvekkel. Hogy lehet, hogy mégis maradt az eredeti kínálat, az angol, a német és a francia?

— Mi minden tőlünk telhetőt megtettünk. Sokat tárgyaltam nemzetiségi kultúrházvezetőkkel, belföldi és külföldi intézetekkel, szervezetekkel, de mint kiderült, nem igénylik a rendszert. Így hát újabb nyelvekre nem is dolgoztuk ki.

— Nem másolják illegálisan a LINGUASOFT-ot? Mi lesz, ha feltörik a programvédelmet?

— Eddig még nem találkoztam kalózpéldánnyal. Ez lehet a védelem érdeme is, de szerintem inkább a terjesztési rendszernek köszönhető. Nincs ugyanis két egyforma LINGUASOFT-hajlékonylemez. Speciális másolóprogramunk van, amely minden egyes másolatra rávisz valami kis eltérést. A lemezeket névre szólóan adjuk el, és a vevő tudomásul veszi, hogy nem másolhatja azokat. Ezt aláírásával is megerősíti. Ha illegális másolattal találkoznánk, nyilvántartásunk alapján meg tudnánk állapítani, kinek a lemezéről készült. Örülök, hogy erre még nem került sor. A nyelvtanárok és programozók igen nagy munkát fektettek a rendszerbe, és biztosan elmenne a kedvük, ha másolni kezdenék szellemi alkotásukat.

— Mikor fejeződött be a rendszer fejlesztése? Újra kezdené-e mai tapasztalataival a munkát?

— A rendszert folyamatosan finomítjuk és tökéletesítjük. A jelenlegit 1987 májusában fejeztük be. Kilenc-tíz programozó és három nyelvtanár dolgozott rajta. Nagyon szeretem a közös művünket, de bánt, hogy sokan lebecsülik a benne lévő teljesítményt. Nem biztos, hogy újra kezdeném.

— Hogyan tudta rávenni a nyelvészeket erre a számukra szokatlan munkára, és mit szóltak a kollégáik?

— Elég nehezen, de sikerült felkelteni az érdeklődésüket azzal, hogy nem könyvben, hanem hajlékonylemezen rögzítjük tudásukat. Gráfszerű sémákat dolgoztunk ki papíron és kértük, hogy töltsék ki azokat. Kollégáik nem nagyon értették meg őket; a LINGUASOFT-ban pedig annyira nem hisznek, hogy meg se nézték.

— Miért nem valamelyik elterjedt vállalkozási formát választotta ahelyett, hogy egy művelődési ház keretein belül működjön?

— Néhány sikertelen próbálkozás után döntöttem így. A művelődési házzal jó a kapcsolatom. A forgalomból részesedik, így érdeke a rendszer terjesztése. Én a Szerzői Jogvédő Hivatalon keresztül kapom a díjazásomat. Ez így nekem teljes mértékben megfelel.

— Térjünk vissza a LINGUASOFT-hoz. Milyen programozási nyelven készült?

— Assembler nyelven, de vannak BASIC részei is. A program felépítése a fordítóprogramokéhoz hasonlít. Az ötletet is egy rendszerprogramozásról szóló könyvből merítettem.

— Vállalatok, szövetkezetek nem érdeklődnek a LINGUASOFT iránt?

— Több vállalattal folytattam megbeszélést, de eddig konkrét üzlet nem jött létre. Egy amerikai céggel ellenben már alá is írtuk az értékesítési szerződést.

— A standjukon PC XT-t is láttam. Ez azt jelenti, hogy a LINGUASOFT PC-re is elkészült?

— Nem, azon a gépen egy levelezési segédlet program fut. A LINGUASOFT-ot ígéretünk ellenére nem készítettük el PC-re, mert nem volt elég nagy az érdeklődés. Abban a körben, ahol ezt a rendszert vásárolják, még nem terjedtek el a PC-k. Elkészítjük azonban a LINGUASOFT alapfokú nyelvi változatát, amely sokkal kisebb és olcsóbb lesz.

— Kazettán is lehet majd kapni?

— Nem. Az a programozási módszer, amivel dolgozunk, nemigen valósítható meg kazettán. Az alapfokú LINGUASOFT után az idegen nyelvű levelezési segédletet fejlesztjük tovább. Ez a rendszer tulajdonképpen mondatkonzerveket tartalmaz, a Datorg hasonló című könyvei alapján. A mondatok fa-szerkezetben lekérdezhetők. Szinte nem is kell hozzá nyelvtudás: a magyar szöveg alapján a programban tárolt mondatkonzervekből, azaz szabvány mondatokból a képernyőn könnyen összeállítható egy üzleti levél. A főmenüből ki kell választani a kívánt témát: üdvözlés, árajánlat, reklamáció stb. Ezután a képernyőn megjelennek a tipusmondatok, amelyek közül kurzorral lehet válogatni. Egyes mondatokban vannak kipontozott részek, például a termék neve számára. A megfelelő szavakat a szótár-alprogramból hívhatja le a levelező, tehát azokat sem kell ismernie. Mind több vállalat kapja meg az önálló külkereskedelmi jogot, de kevés az idegen nyelvű levelező. Ezért is dolgoztuk ki ezt a segédletet, egyelőre angol, német, francia, spanyol és magyar nyelvre.

— Magyarra is? Vagyis külföldiek is használhatják, magyar nyelvű levelek megírására?

— Igen. A program működési elve ugyanaz, csak a menü és a lekérdezési fa-struktúra az adott ország nyelvén készül. A külföldi forgalmazással a Datorg foglalkozik. Úgy tudom, néhány üzlet már létre is jött.

Rövid beszélgetésünk végén azt kívánom Gálai Antalnak, hogy ne veszítse el a kedvét, hanem ezután is minél gyakrabban hökkentsen meg bennünket újabb ötleteivel. — p —

Mikrokalauz

Történetesen, ha például Archimedes ma merülne fürdővizébe, bizony jó kétezer éves késésben lennénk önmagunkhoz képest. Az imigyen visszafordíthatatlan és tornyosuló lemaradásunkon az sem segítene, ha pusztán fürösztéssel igyekeznénk felfedezni ama archimedesi törvényt. Vajmi kevesen jönnénk rá. Szerencsénkre Archimedes azon kevesek közé tartozott, akik idejében vettek fürdőt. Noha esetlegesnek tűnik a tudománytörténeti példa, rejt magában egy nagyon fontos praktikus szempontot. A felfedező és a laikus közötti különbséget. A felfedező mindig pontos, zsenialitása szinte időtlen. A laikus viszont kullog utána, s egyáltalán nem mindegy, mikor, honnan szerez a heurékáról tudomást, mikor fedezi fel magának.

A hazai számítástechnikában e gond szembeötlő. Már a profi számítástechnikusoknak is nehéz a világ innovációs fejlődésének gyorsaságával ilyetén lépést tartaniuk. A nem számítástechnikus szakembereknek, a fentiek szerint úgymond laikusoknak, miközben egyáltalán nem szükséges a tudomány és technika XX. századi lenyűgöző dómját aprólékosan megismerni, munkájukban mind közelebb kerülnek hozzá. Munkaeszköznek a mikroszámítógépet választják. Igen, de éppen a választás pillanata a leggyötrelmesebb, mert hogy mit vegyen, honnan, ehhez talál legkevesebb információt. Éppen itt éri hidegzuhany. Hiszen ez az a köztes helyzet, amikor a szakismeretre még nem kell képeznie magát, amihez jószerivel fellelhet már kellő irodalmat, ellenben a tájékozottság nélkülözhetetlen. Ám — Si animus est, non deest auxilium: Hogyha van lelkesedés, nem hiányzik a segítség sem. Ez utóbbira vállalkoztak a Mikrokalauz szerzői a MESZ Számítástechnika kebelében.

A Mikrokalauz legnagyobb erénye, hogy naprakész. Azaz folyamatosan megújul a világból és az itthoni számítástechnikai információs forrásokra támaszkodva. A Mikrokalauz ugyanakkor sorozat, az egyes kötetek felölelik a hazai mikroszámítógép- és szoftverpiacot, a számítástechnikai irodalmat, s mindenekelőtt az első kötet betekintést nyújt a mikro-számítástechnikai alapokba és az egész iparág fejlődési tendenciáiba.

Könnyelműen és talán túlontúl pragmatikusan, a vásárló önmagát sürgetve elhamarkodottan úgy ítélheti, erre a kötetre egyáltalán nincs is szüksége. De gondoljon mindenki egy pillanatra mondjuk egy televízió vásárlására. Bármennyire egyszerű és közhelyszerű az arról való ismeret, mégis érdeklődik az iránt, mit tud egy korábbi típushoz képest, mióta gyártják, és így tovább. Hát még akkor milyen fontos a mikroszámítógépekről vásárlás előtt egyet s mást tudni? Márpedig az első kötet a számítástechnika aktuális helyzetét körvonalazva ehhez nyújt kitűnő segítséget.

Az alapokról és fejlődésről írva a dolog természeténél fogva nem kerülhető meg a történetiség. Ez azonban a számítástechnika és ezen belül a mikroelektronika bemutatásában kifejezetten jótékony hatású, mert mint azt egy ókori filozófus mondotta: minden tudás emlékezés csupán.

A Morzsától a mikrogépig fejezet, az alapvető fogalmak tisztázásán kívül, rögtön bőkezűen ontja a legsokoldalúbb fejlődési irányokat. Így például a világon a mikroprocesszorok gyártásának, illetve a mikroszámítógép forgalmának várható alakulását 1990-ig. Megtudhatjuk azt is, hogy az európai szocialista országok majd' mindegyike gyárt már 8 bites mikroprocesszort és a Szovjetunió, az NDK, valamint Csehszlovákia már a 16 bites mikroprocesszorok gyártását is megkezdte. A leendő felhasználót egyben eligazítja abban is, amikor a mohósága esetleg feleslegesen verné horribilis összegekbe, mert hogy: „Olyan viharos gyorsaságú kategóriaváltásra a 16 és 32 bites között nem kell számítani, mint amilyen a 8 és 16 bites gépek csatájában történt. A magyarázat igen egyszerű: a 16 bites gépek teljesítménye már messzemenően kielégíti a mindennapos felhasználói igényeket az alkalmazói tömegek esetében. Így például az irodai alkalmazásnál, s egyszerűen nincs kényszerítő motivációja a 32 bites gépekre való áttérésnek. Ez a kényszerpálya csak igazán számításigényes alkalmazásoknál merül fel, mint a képfeldolgozásnál, a grafikánál, a tervezői munka támogatásánál, sokterminálos (10 fölötti) lokális hálózatoknál. Tehát aki ma IBM PC-vel kompatibilis gépet vásárol, az hosszú távon, legalább 10 évig korszerű bázison, gazdag, hatékony szoftverállománnyal alkalmazhatja a számítástechnikát. Így közvetve részese lehet annak a termékfelruházásnak, amellyel a világ legnagyobb számítástechnikai cége, az IBM, amely a maga 370 ezer dolgozójával hazánk éves nemzeti jövedelmével megegyező termelési értéket hoz létre évente, jelenleg kétségtelenül a központi termékét adja."

A gép és a külvilág közötti kapcsolatot teremtő és tartó perifériákról szóló fejezetben a már nekilendült kíváncsiságot tovább csigázzák a szerzők. Nem maradnak itt sem adósak persze a háttértárak, nyomtatók típusonkénti bemutatásával. „A farkasétvágyat e körben is azonban a jövőt jelentő inycsiklandozó falatokkal gerjesztik. Talán a legizgalmasabb a Bernoulli Box elnevezésű tároló, mely nevét Daniel Bernoulli svájci matematikusról kapta. Ez az új tárolóeszköz rendkívül gyorsan terjed az Egyesült Államokban. 1986 tavaszán már több mint kétszázezer működött belőle. A kifejezetten mikrogépekhez szánt tároló egyesíti magában a hajlékonylemez hordozhatóságát, valamint a merevlemez kapacitását és sebességét. A legújabb modelljének méretei: 320 × 18 × 432 mm, s ez a doboz tartalmaz egy vagy két félmegás meghajtót, tápegységet és hűtőventilátort. A meghajtók kapacitása egyenként 10 vagy 20 megabájt, így tehát 20-40 megabájt tárolási kapacitást kínál egyetlen egység. A 40 megabájtos változat ára 1986-ban Angliában 3500 font volt. A Bernoulli lemez egy kemény, hozzávetőlegesen A/4 méretű kazettába van zárva, maga a lemez hajlékony, átmérője 8 hüvelyk. Ezt a meghajtó 3000 fordulat/perc sebességgel forgatja, ami a mágneses hajlékonylemezes tároló sebességének a tízszerese. A Bernoulli Box vonatkozásában jóslatokba ma még korai bocsátkozni, de a paraméterei alapján reális konkurenciát jelent a winchester táraknak." És vajon mikor jönnek el nálunk azok a boldog idők?!

S hogy akad a tempón mit gyorsítani, azt mi sem mutatja jobban, mint az adathordozó lemezek gyártásának summája egy rövid összevetésben. A világon alig féltucatnyi cég gyárt hajlékonylemez-alapanyagot, s azt tekercsekben szállítja a gyárak többségéhez, melyek kiszerelik ezeket. A szocialista országokban egyelőre még nem gyártanak hajlékonylemez-alapanyagot. Kiszereléssel is csak a bulgáriai IZOT cég foglalkozik.

A számítógépes rendszerből a lélek sem maradhat ki. A szoftverrel pedig nem lehet lélektelenül foglalkozni. Ezért is emelik ki a szerzők, hogy: „Végül is az alkalmazói programok döntik el, hogy mennyire hatékonyan elégíti ki a számítógép az emberi szükségleteket. Ezért valószínű, hogy a személyi számítógép tulajdonosa többet fog szoftverre fordítani, mint hardverre. Költhet úgy, hogy programokat vásárol, de úgy is, hogy jelentős mennyiségű időt fordít a megírásukra. Hacsak nem akar valaki intenzíven programozni, a hardver kiválasztásánál komolyan figyelembe kell vennie a rendszer szoftverbázisának szélességét és mélységét, azaz az egyes alkalmazásokhoz kapható különböző programok számát."

Ezzel a gondolattal azonban máris két nagyobb témához érkezünk, a hazai mikroszámítógép- és szoftverpiachoz. De erről majd később írunk. Mindazonáltal maradjunk egy percre még az első kötet jelentőségénél, méltatásánál. Ha valakinek eltökélt szándéka a személyi számítógép használata, akkor igényeinek megfontolásához pontos iránytű a Mikrokalauz, habár bakot lőhet az is, aki egyébként jól céloz.

Szulyovszky Csaba

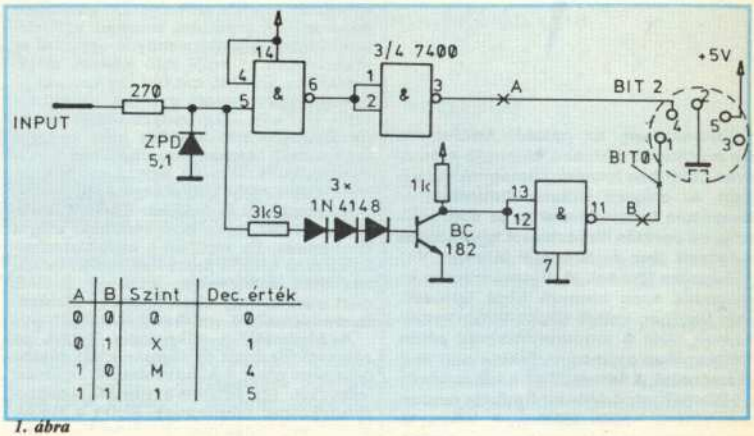
Logikai szintjelző

A digitális technikával foglalkozók fontos segédeszköze a logikai szintjelző. Ez általában egy ceruza formájú, a hegyénél tűszerű tapintóval ellátott eszköz, amely a tapintóval megérintett áramköri pont logikai állapotát mutatja.

Az állapotok a digitális technikában megszokott elnevezésekkel LOW (L=alacsony), HIGH (H=magas), MIDDLE (M=közepes) állandó szintek vagy gyorsan változó alacsony-magas szintű jelsorozatok, PULSE (P=impulzus) lehetnek.

A különféle állapotok különféle színű LED-ekkel jeleníthetők meg. A bonyolultabb szintjelzőknek tárolójuk is van, amely képes egy előző állapotot megőrizni. A szintjelzők áramköri szempontból nem túl bonyolult eszközök és nem is nagyon drágák.

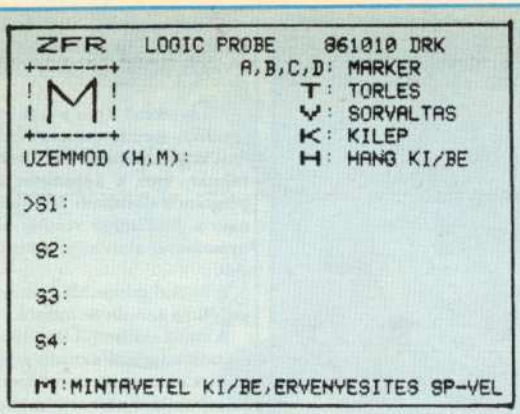
A következőkben bemutatott, személyi számítógép bemenetére kapcsolódó szintjelző kitűnő illusztrációja annak, hogyan lehet ezt az egyszerű eszközt — alapvető funkciójának megtartásával — intelligens



1. ábra

```
8000  CD C9 01 FD 21 BC 02 21 F9 82 22 4B 40 21 E1 81   ....!..!.."K@!..
8010  CD 75 2B 3E 06 CD 15 00 CD 5E 81 3E 80 CD 6D 81   .u+>.....^.>..m.
8020  0E 04 06 00 CB 40 28 32 CB 48 28 02 18 05 79 FE   .....@(2.H(...y.
8030  04 28 27 79 C5 01 0A 00 FE 05 28 0D FE 00 28 0E   .('y......(...(.
8040  FE 02 28 0F 11 00 03 18 0D 11 00 01 18 08 11 00   ..(.............
8050  05 18 03 11 00 04 CD 68 3F C1 CD 1D 00 CA FF 80   .......h?.......
8060  F5 CD 1D 00 20 FB F1 FE 20 28 16 FE 54 28 91 FE   .... ... (..T(..
8070  56 28 38 FE 4B 28 2C FE 48 28 58 FE 4D 28 66 18   V(8.K(,.H(X.M(f.
8080  06 CB 48 20 7F 18 D3 D6 41 DA FF 80 FE 04 38 03   ..H ...A.....8.
8090  C3 FF 80 C6 41 CD 9B 81 3E 20 CD 9B 81 CD 4B 81   ....A...> ....K.
80A0  C3 FF 80 3E 16 CD 15 00 C3 00 00 FD E5 D1 CD 80   ...>............
80B0  00 3E 20 CD 15 00 14 14 7A FE 0F 20 02 16 07 D5   .> ....z.. ....
80C0  FD E1 CD 80 00 3E 3E CD 15 00 CD 5E 81 CD 4B 81   .....>>....^..K.
80D0  C3 FF 80 1E 10 CB 40 28 06 CB 80 3E 20 18 16 CB   ......@(...> ...
80E0  C0 3E 48 18 10 1E 0C CB 48 28 06 CB 88 3E 20 18   .>H.....H(...> .
80F0  04 CB C8 3E 4D 16 05 F5 CD 80 00 F1 CD 15 00 CB   ...>M...........

8100  48 C2 5A 80 CD AF 81 CB 48 20 05 FE 00 CA 24 80   H.Z.....H ....$.
8110  79 FE 05 28 0D FE 00 28 0E FE 02 28 0F 21 4D 80   y..(...(...(.!M.
8120  18 0D 21 31 84 18 08 21 30 82 18 03 21 50 86 7C   ..!1...!0...!P.|
8130  CD 6D 81 CB 48 20 05 FE 80 CA 24 80 7D CD 9B 81   .m..H ....$.}...
8140  3E 20 CD 9B 81 CD 4B 81 C3 24 80 C5 FD E5 D1 3E   > ....K..$.....>
8150  04 83 5F CD 80 00 21 39 83 CD 75 2B C1 C9 C5 06   .._...!9..u+....
8160  1B 21 39 83 36 20 23 10 FB 36 00 C1 C9 F5 3E 02   .!9.6 #..6....>.
8170  CD 15 00 11 04 02 CD 80 00 3E 20 CD 15 00 11 30   .........> ....0
8180  01 CD 80 00 3E 20 CD 15 00 11 CC 00 CD 80 00 F1   ....> ..........
8190  CD 15 00 F5 3E 0C CD 15 00 F1 C9 21 39 83 C5 01   ....>......!9...
81A0  25 00 09 23 E5 D1 2B 36 00 ED B8 23 77 C1 C9 DB   %..#..+6...#w...
81B0  40 E6 05 57 5F C5 06 7F DB 40 E6 05 A2 57 DB 40   @..W_....@...W.@
81C0  E6 05 B3 5F 10 F2 C1 7A BB 20 06 B9 28 10 4F 3C   ..._...z. ..(.O<
81D0  C9 7A AB FE 05 20 D8 3E 02 B9 28 02 4F C9 3E 00   .z... .>..(.O.>.
81E0  C9 02 20 5A 46 52 20 12 4C 4F 47 49 43 20 50 52   .. ZFR .LOGIC PR
81F0  4F 42 45 20 20 20 38 36 31 30 31 30 31 20 44 52   OBE   8610101 DR
```



2. ábra

tulajdonságokkal felruházni, használhatóságát kiterjeszteni.

Először is a számítógépbe kell vinni az áramkör állapotát meghatározó jeleket. Mivel három különböző jelszint van (H, M, L), ezt az információt legalább két bit segítségével különböztethetjük meg.

Ezután el kell dönteni, hogy a számítógépbe bevitt jelekkel egy időben a szintjelzőn is megjelenjen-e az információ, tehát hogy önállóan, a számítógéptől függetlenül is használható legyen-e az eszköz. Ez mérőműszerek esetében annak felel meg, hogy a műszer önállóan is kezelhető, beállítható, a mért érték leolvasható, de a műszer számítógéppel is vezérelhető. Terjedőben van egy olyan irányzat, hogy a mérőműszereknél ezt a kettős funkciót megszüntessék: a műszerek előlapján levő — sokszor bonyolult kialakítású és meglehetősen költséges — kezelő- és kijelzőszerveket elhagyják, és a műszert kizárólag számítógép által vezérelt működésre alakítják ki. Ennek az az előnye, hogy sokkal bonyolultabb funkciók is könnyen megvalósíthatók a programozhatóság kihasználásával — nem kell millió billentyűt és azok megfelelő sorrendű megnyomását megtanulni, hanem a képernyőn megjelenő részletes információ alapján választhatunk. Egyszerűbb a műszer kisebb megbízhatóságú mechanikai részeinek kialakítása is. A kettős funkcióról való lemondás megfelelő gyártási környezetben olcsóbbá teszi a műszert. Esetünkben is célszerű lemondani a kettős megjelenítés lehetőségéről, mert ezáltal az áramkör egyszerűbbé válik.

A bemeneti logikai szinteket két bitre kódoló áramkört az *1. ábra* mutatja.

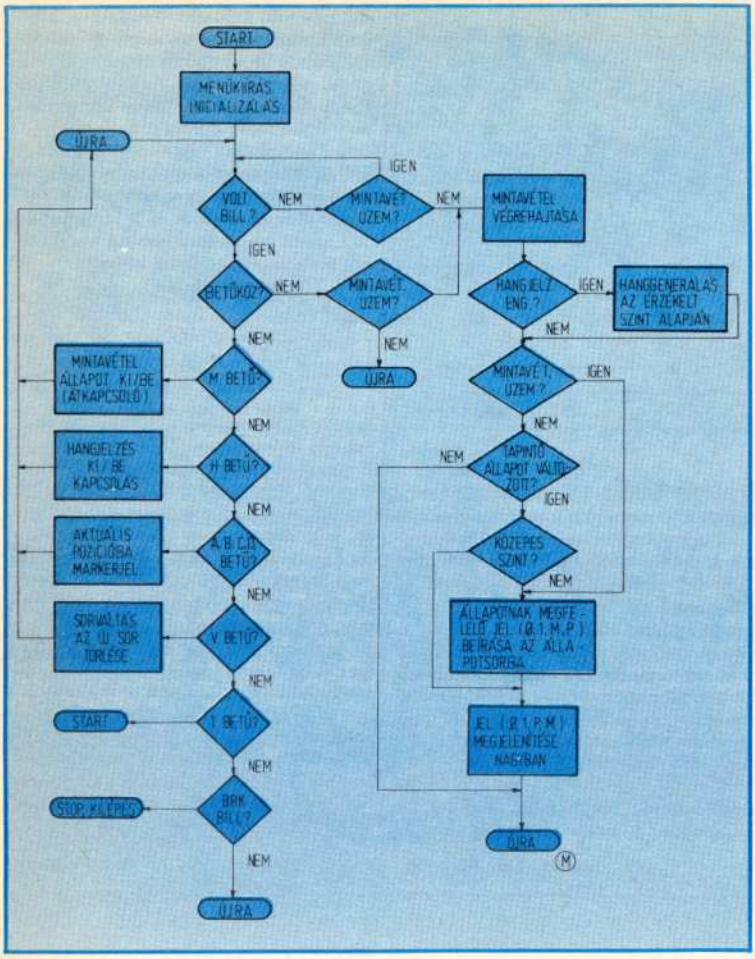
A mindössze egy SN74ØØ vagy SN74LSØØ típusú integrált áramkört tartalmazó áramkör a bemenetére kapcsolt jel szintjétől függően az A és B kimeneteken TTL szintű jeleket ad ki, melyek a számítógép két bemeneti pontjára köthetők:

Jelszint a tapogatón	A kimenetek	B
alacsony (L vagy 0)	0	0
közép (M)	1	0
magas (H vagy 1)	1	1

A következő probléma a beolvasott jelek (a két bit) feldolgozása. Mivel a szintjelzővel folyamatosan dolgozunk, a jeleket a számítógépbe folyamatosan kell beolvasni és megjeleníteni. Egy megoldási mód az lehetne, hogy minden beolvasás után megjelenítés következne. Ez azonban azért nem jó, mert olyan gyors, villogó kijelzési kép keletkezne, amely nagyon fárasztaná a szemet. A helyes megoldáshoz induljunk ki a gyakorlatból. Használat közben a tapogató igen sokszor nem kapcsolódik semmihez, a „levegőben van", és ilyenkor a kimeneteken az 10 (M) kombináció van. Ha a tapogatót a vizsgált logikai hálózat egy pontjához érintjük, az a következő állapotokat jelezheti: alacsony (L), közepes (M), magas (H) szint vagy alacsony és magas szintek gyors változása, azaz impulzusok (P).

```
8200  4B 20 0D 20 2B 2D 2D 2D 2D 2D 2D 2B 20 20 20 20   K . +-------+
8210  20 20 20 20 20 20 41 2C 42 2C 43 2C 44 3A 20 4D       A,B,C,D: M
8220  41 52 4B 45 52 0D 20 21 20 20 20 20 20 20 21 20   ARKER. !       !
8230  20 20 20 20 20 02 20 20 20 20 54 12 3A 20 54 4F        .    T.: TO
8240  52 4C 45 53 0D 21 20 20 20 20 20 20 20 21 20 20   RLES.!       !
8250  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
8260  20 20 20 02 56 12 3A 20 53 4F 52 56 41 4C 54 41      .V.: SORVALTA
8270  53 20 0D 20 2B 2D 2D 2D 2D 2D 2D 2B 20 20 20 20   S . +-------+
8280  20 20 20 02 20 4B 12 3A 20 4B 49 4C 45 50 0D 20      . K.: KILEP.
8290  55 5A 45 4D 4D 4F 44 20 28 48 2C 4D 29 3A 20 20   UZEMMOD (H,M):
82A0  20 20 20 20 20 02 48 12 3A 20 48 41 4E 47 20 4B       .H.: HANG K
82B0  49 2F 42 45 0D 0D 3E 53 31 3A 0D 0D 20 53 32 3A   I/BE..>S1:.. S2:
82C0  0D 0D 20 53 33 3A 0D 0D 20 53 34 3A 0D 0D 02 4D   .. S3:.. S4:...M
82D0  12 20 3A 20 4D 49 4E 54 41 56 45 54 45 4C 20 4B   . : MINTAVETEL K
82E0  49 2F 42 45 2C 45 52 56 45 4E 59 45 53 49 54 45   I/BE,ERVENYESITE
82F0  53 20 53 50 2D 56 45 4C 00 42 66 5A 5A 5A 5A 5A   S SP-VEL.BfZZZZZ

8300  42 42 42 42 42 42 42 42 43 7E 42 42 42 42 42 46   BBBBBBBBC~BBBBBF
8310  4A 52 62 42 42 42 42 42 7F 02 06 0A 12 22 42 02   JRbBBBBB....."B.
8320  02 02 02 02 02 02 02 02 03 7E 42 42 42 42 42 42   .........~BBBBBB
8330  42 42 7E 40 40 40 40 41 00 00 00 00 00 00 00 00   BB~@@@@@A.......
8340  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
8350  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
```



3. ábra

Ezért a jó megoldás az, hogy amíg a tapogatón lévő jelszint nem változik, addig az állapotot csak egyszer jelezzük ki. Változáskor, annak jellegét figyelembe véve történik az újabb kijelzés.

A működési algoritmus a következő. Sokszor, például százszor egymás után beolvassuk a logikai állapotot reprezentáló két bitet. Ha ez minden egyes beolvasáskor azonos, a bemeneten statikus, állandó jelszint van. Ha a beolvasott sorozat nem azonos, akkor két eset lehetséges:

1. A beolvasás során a tapogatót levettük (vagy ráraktuk) a vizsgált pontról (pontra) és a bemeneten L—M, H—M, M—L, M—H átmeneti változások voltak, ami a tapintó mozgatására utal. Újabb beolvasást kell végrehajtani, egészen addig, amíg vagy statikus szintet, vagy impulzussorozatot nem érzékelünk.

2. L—H, H—L átmenetek érzékelése azt jelenti, hogy a bemeneten impulzussorozat van.

Ezek után azt kell eldönteni, hogy az így kiértékelt állapot az előző beolvasássorozat állapotával megegyezik-e. Ha igen, akkor az állapotot nem kell megjeleníteni, mert az már előzőleg megtörtént. Különben ki kell jelezni, vagy a tapintó mozgatására utaló kiértékelés esetén a beolvasást — kijelzés nélkül — meg kell ismételni.

A beolvasott jelsorozat kiértékeléséhez azt a tényt használjuk fel, hogy ha a beolvasás során valamelyik bit értéke megváltozik, akkor ez két logikai művelet, az ÉS és VAGY műveletek bemeneti bitekre való alkalmazásával meghatározható.

A beolvasott első érték és a további értékek közötti ÉS kapcsolat az 1—0 változást, a VAGY kapcsolat a 0—1 változást mutatja ki.

A következő lépés a működési mód és a funkciók megtervezése. A modern programok döntő többsége ún. menütechnikát alkalmaz, azaz a képernyőn megjeleníti a program választható funkcióit. A felhasználó a funkcióhoz rendelt billentyű megnyomásával aktivizálja a kiválasztott funkciót.

A logikai szintjelzőhöz megtervezett képernyőkép a *2. ábrán* látható.

A menü mellett jól láthatóan meg kell jeleníteni a tapintó aktuális állapotára vonatkozó karaktert is: ez a képernyő bal felső sarkában lévő négyzetben látszik. A képernyőn a tapintó előző állapotai az ún. állapotsorba kerülnek, az új állapot megjelenésekor a sor eggyel jobbra lép.

A képernyőn négy állapotsor található (S1—S4), amelyekből mindig egy aktuálisat választhatunk ki a V billentyű segítségével, és ezt a > jel jelöli.

Az A, B, C, D karakterek leütésével az állapotsorban jelet (markert) helyezhetünk el. A leütött karakter megjelenik az állapotsorban, például megjelölve egy adott áramköri rész vizsgálatának kezdetét.

Megoldható a tapintó állapotát jelző különféle magasságú hangjelzés generálása is (a H billentyűt nyomva bekapcsolja a hangjelzést, ismételt megnyomásra kikapcsolja.)

Mivel a közepes (M) szint nem kerül az állapotsorba, ezért ha például egy integrált áramkör minden lábát letapogatjuk, akkor az állapotsorban nem jelennek meg az ilyen állapotú lábakhoz tartozó értékek, vagyis az állapotsor nem képezi le teljesen a kivezetéseket. Ez is megoldható azonban, ha a tapintó állapotát nem folyamatosan, hanem egy adott parancsra (a betűkhöz billentyű megnyomására) mintavételezzük, és utána írjuk az eredményt az állapotsorba. Ez az üzemmód az M billentyű megnyomásával kapcsolható ki és be.

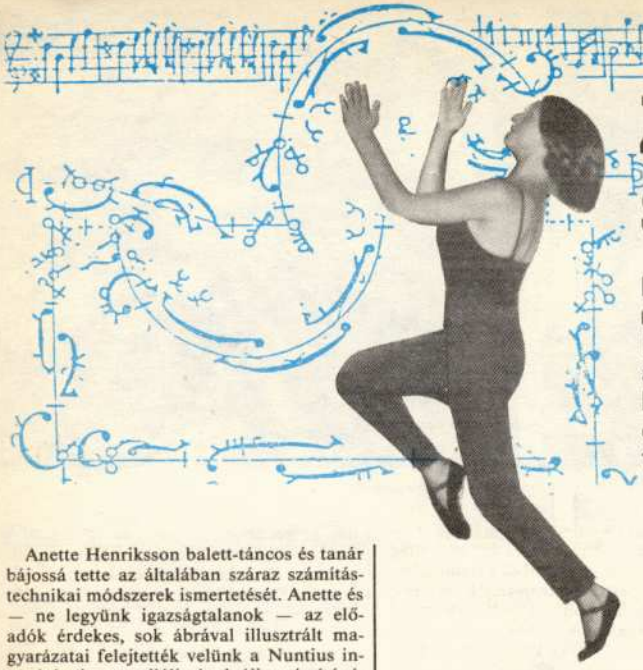
A fentieket megvalósító program blokkvázlatát a *3. ábrán* láthatjuk.

A blokkvázlat alapján bárki elkészítheti saját számítógépére a programot, akár BASIC-ben, akár gépi kódban.

Mivel a logikai szintjelző eredeti, működő változata Primóra készült, a Primo-tulajdonosok számára közöljük a gépi kódú program hexadecimális listáját. A program a 8000H címről indítható (BASIC-ből: ?CALL(—32768)).

Megjegyzés: Az 1. ábrán látható kapcsolású szerelt és működő logikai szintjelző áramkör a szerkesztőségben keresztül, írásban megrendelhető. Ára: 150,— Ft+utánvét. Megrendelhető a Primo-program önindulós, kazettán rögzített változata is. Ára: 100,— Ft+utánvét.

DR. KÓNYA LÁSZLÓ



Zene és tánc számítógépen

Izgalmas eseményre érkezett meghívó szerkesztőségünkbe: Rajka Péter és Ungvári Tamás, az Electronic Music in Sweden kutatóközpont munkatársai tartanak előadást; a programot video- és táncbemutató színesíti.

Anette Henriksson balett-táncos és tanár bájossá tette az általában száraz számítástechnikai módszerek ismertetését. Anette és — ne legyünk igazságtalanok — az előadók érdekes, sok ábrával illusztrált magyarázatai felejtették velünk a Nuntius interaktív, intermediális és duális számítógépes rendszer ijesztő hangzását, ezen belül a Motográphikont és a Macrosticont.

Mik is ezek a rejtélyesen hangzó dolgok? A Motográphikon egy mozgásjegyzési, analizálási és animációs számítástechnikai, a Macrosticon pedig egy jelfeldolgozó, szintetizáló és megjelenítő zenei rendszer, szintén számítógépen.

Még egy fogalmat kell előrebocsátanunk: a notációt. Jelen esetben ez a zene és tánc (balett) feljegyzése.

A zene a művészetnek az az ága, amelyben az alkotás lényegét hangok különféle kombinációi fejezik ki. A hangok időbeli egymásutánja dallamot alkot, egyidejű megszólalásukból pedig harmónia keletkezik. Mozgató ereje a ritmus, sebessége a tempó. A balett több száz éves hagyományon alapuló színpadi tánctechnika: 1662-ben Franciaországban foglalták a szabályait rendszerbe.

Jogos a kérdés: mi köze e két művészeti ágnak a számítástechnikához? Amikor az NJSZT meghívóját kézhez kaptuk, minket is ezek a kérdések foglalkoztattak. Rajka Péter és Ungvári Tamás hamar eloszlatta a kételyeinket. Ungvári Tamás elmondta, hogy a zenei notáció viszonylag hamar kialakult a kottázás (hangjegyírás) formájában, de több olyan hiányossága van, amely a zene tánchoz való rendelését is nehezíti.

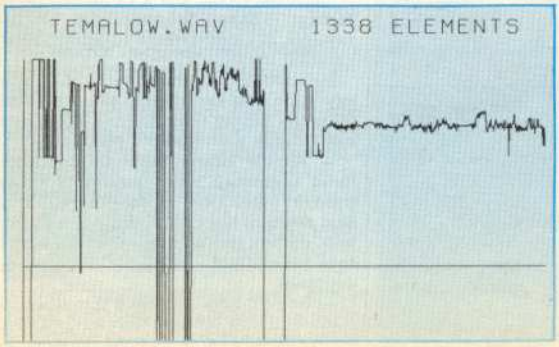
A számítógéppel komponált zenével kapcsolatban az egyik fő ellenvetés az, hogy a nemzetközi szerzői jogvédelem még nem fogadja el az ezidáig lehetséges dokumentálási formákat (magnófelvétel, programlista), és így nem lehet dokumentálni; védeni sem tudják ezeket az alkotásokat.

Ungvári Tamás egy DEC PDP—15 modellel, illetve VAX—11/750 rendszerrel dolgozik, komponál. A rendszer lelke az FP 120B tömbprocesszor. Ungvári Tamásnak a zenekritikusok és a hallgatóság véleménye szerint is valóban humanista — nem kemény és nem embertelen — zenéjét meghallgathattuk. Művészeti munkásságán túl a többségében szakmánkbeli közönséget megragadta a művészember számítástechnikai teljesítménye: kialakította a számítógépen komponált zene notációs rendszerét. Az általa megálmodott és számítógépen megvalósított hangzások megjelennek a papíron szonogram és spektogram formájában (*1.* és *2. ábra*).

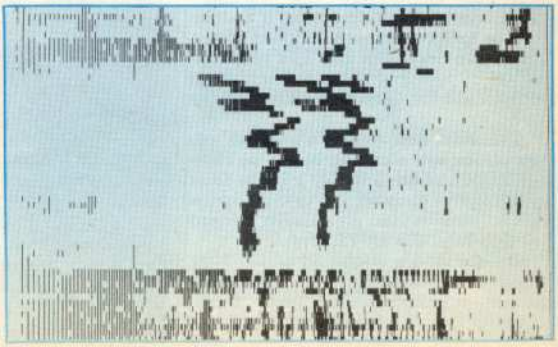
A szonogram esetében a gép által nyomtatott koordinátarendszerben az egyik tengelyen a hang frekvenciáját, a másikon az időt látjuk. Ez már majdnem tökéletes lejegyzése a zenei műnek. A spektogramon pedig a fentieken kívül a hangzás erőssége is látható, az árnyalatok által.

A Motográphikon a tánc lejegyzési rendszerét forradalmasíthatja, de még ennél is többet jelent. Rajka Péter koreográfus a tánc notációjában hasznosítja a modern technikát. A tánc mozdulatainak lejegyzése sok évszázados igény. A bemutatott jelenetek reprodukálása már a középkori Kínát is foglalkoztatta. Így több notációs rendszer

1. ábra



2. ábra



Új koreográfia születik

Együtt a művészpáros

született. A *3. ábra* egy ilyen próbálkozást szemléltet. A ma használt rendszereknek is fő hiányosságuk, hogy nem jelölik az időt, és így a zenével való kapcsolatot sem lehet egyértelműen leolvasni. Ez az alapvető igény késztette az új notációs eljárás kidolgozását. Az első lépés a szimbolikus rendszer kialakítása volt. Ezáltal a térben a jelek, az időben a függőleges tengely igazítanak el. A *4. ábra* a szimbólumok és a mozgás kapcsolatát mutatja be. A jeleket a számítógépbe egérrel vagy billentyűzetről lehet bevinni.

Megkérdeztük Rajka Pétert, hogy a videotechnika korában miért van szükség a mozdulatok lejegyzésére. Elmondta, hogy a filmen látható kép nem ad egzakt lehetőséget a táncosok elkülönülő mozdulatainak térbeli reprodukálására, de a legnagyobb baj az, hogy a nézőben, a koreográfusban óhatatlanul szubjektív érzéseket kelt. A papíron vagy számítógéppel lejegyzett mozdulatok ellenben a „vegytiszta" mozdulatokat adják vissza.

Az általa Macintosh számítógépre írt program lehetőséget teremt a tánc notációjára, a mozdulatok elemzésére, sőt az animációs rész egy kis figura alakjában be is mutatja a művet a képernyőn.

Anette Henriksson az előadás után elmesélte, hogy az illusztrációként bemutatott balett koreográfiájára már nem is emlékezett, de hajlékonylemezen elhozott lejegyzés alapján negyedóra alatt újra megtanulta. Sőt, a Motográphikon segítségével új mozgáselemeket, kombinációkat fedezett fel.

A rendszer nem csupán a meglévő művek notációjára alkalmas, hanem új koreográfia készítésére is — így a gépen lehet megtervezni a mozgást. Van egy segédprogram is, amely a tánc dinamikáját kifejező jelleggörbéket állít elő (*5. ábra*).

Különös módon vethető fel ezután, hogy a tyúk volt-e előbb vagy a tojás? Ugyanis a művész—számítástechnikus páros bemutatkozása előtt az volt a helyzet, hogy szinte kizárólagosan a zenei mű született meg előbb, és utána a balett koreográfiája. Most már egy kicsit a feje tetejére állhat a világ!

A szimbolikus rendszerből készült jelleggörbék a zeneszerzőt inspirálják, orientálják, mivel látja, munkája során eleve számolhat a hangzás és a tánc együttes dinamikájával. A művészek között újfajta kommunikációs rendszer alakul ki, mivel a számítógépes zene jelrendszere is visszahat a koreográfus munkájára. Az 5. ábrán olvasóink tanulmányozhatják a két művészeti ág alkotásainak elvonatkoztatott jelrendszerben való közeledését, összhangját.

A bemutató után interjút készítettünk az előadókkal igazán érdekes pályafutásukról.

Nagybőgőtől a számítógépig

Ungvári Tamás 1936-ban született Kalocsán. Hamarosan Budapestre költöztek. A József Attila Gimnáziumban érettségizett. Középiskolás korában elkezdett hegedűn és gordonon játszani, és egyébként is hamar kiütközött alapvető humán beállítottsága. Ez a későbbiekben is végig befolyásolta életét. Ebben az időszakban például megszervezte Budapest összes iskolájának szavalóversenyét.

3. ábra

Hubay-növendékként folytatta a hegedülést, de korábbról hozott rossz beidegződése komoly nehézségeket jelentett — meg is akadt egy időre ez a vonal. Érettségi után a Budapesti Lenfonógyárban dolgozott, mint tűjavító munkás. Erre úgy emlékszik vissza, mint élete egyik legszebb időszakára. Mégis visszahúzta hajlama a művészethez: beiratkozott a Zeneművészeti Főiskolára. 1956-ban a Fővárosi Operettszínház rendes tagja lett, 1957-től az Állami Hangverseny Zenekarban játszott. A tanárképzőt is elkezdte, de saját bevallása szerint nem volt kitartása hozzá.

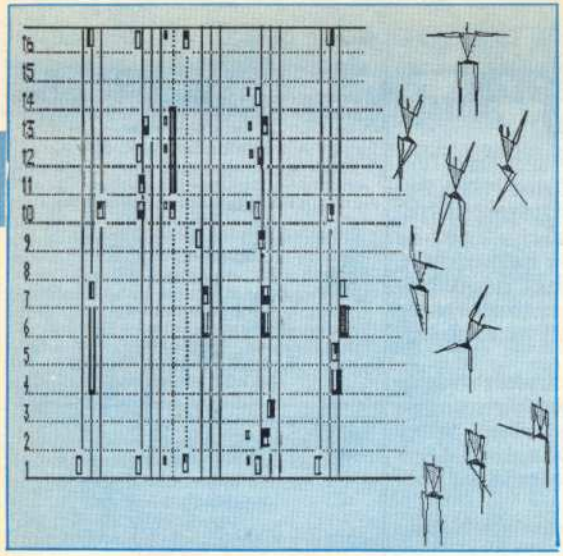
— A filozófia mindig is érdekelt — mondta —, de abban az időben szinte megszállottként foglalkoztam vele. Folyvást újat és újat akartam. Menekülni a konvenciótól! A rutintól való elszakadást kerestem, és a látszólag összefüggéstelen részekben az összefüggést. Nem azt tartottam fontosnak, amin vagy amivel dolgozom, hanem a mögöttes tartalmakat.

1960-tól a salzburgi zenei főiskolán tanul és dolgozik. Ekkor már a karmesteri pálya vonzza. 1970-ben Svédországban köt házasságot, és karmesterként különböző zenekarokkal lép fel.

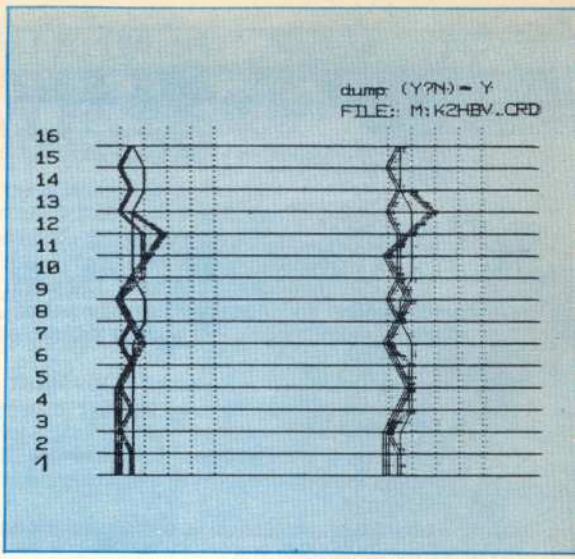
Firenzében karmesterversenyt nyer ugyan, de nem kap állandó állást. Alkalmi fellépései mindig sikert hoznak, de ez nem elég. Megint a továbblépés útját keresi.

— Amikor először hallottam szintetizátoron komponált zenét, megfogott annak iszonyú potenciálja és lehetőségei. Még nem éreztem, hogy zeneszerzőnek kell lennem, de nem tudtam megszabadulni a gépi zene hangzásától — magyarázta pályamódosítását.

— Semmit nem értettem a számítástechnikához. Őrült butának éreztem magam, amikor bejutottam a stúdióba. Programré-



4. ábra



5. ábra

szeket vittem haza, és egyfolytában tanulmányoztam azokat. Néztem, néztem és egyszer csak összeállt.

— Milyen viszonyban van a programozókkal?

— Elég feszült a kapcsolatunk. Általános probléma, hogy lenézik a humán érdeklődésűeket. Az is gyakran előfordul, hogy ha ismerik a programot, már valódi zeneszerzőnek érzik magukat. Pedig a tökéletes hangzásokat, a dinamikai arányokat nem lehet előre „beprogramozni". Ehhez már a művész ihlete kell.

Ungvári Tamást most is elsősorban a filozófia és nem a matematika érdekli. Szép karrier, hogy ennek ellenére a Svéd Zeneakusztikai Intézet számítástechnikai osztályát vezeti.

Koreográfia és kibernetika

Rajka Péter 1949-ben Budapesten született. A Kossuth Zsuzsa Gimnáziumban latin tagozaton végzett. Édesapja freudista, pszichoanalitikus volt, és a családban szinte a levegő alkotóelemeként vette őt körül a humán műveltség. Egész fiatalon, gimnazista korában már koreográfiákat írt.

— Milyen meggondolásból kezdett a tánc írásrendszerével foglalkozni? — kérdeztük.

— Teljesen ösztönösen. Érdekelt a mozgás kódolása. Először a zenei kompozíciós elveket tekintettem irányadónak, később a hagyományos lejegyzési rendszereket tanulmányoztam. Bár általában a legjobbnak tartják és ezért világszerte elterjedt a Laban-féle notációs rendszer, nekem hiányzott belőle a kibernetikai megközelítés lehetősége.

17 éves korában balettozni tanult, és nem sokkal később már a Tháliában és a Vígszínházban táncolt. Katonáskodása alatt is a mozgás struktúrája és a kifejezésmód összhangja foglalkoztatta. Leszerelés után azonban nem találta meg a folytatás általa igényelt lehetőségét.

— Ha jól tudom, ebben az időszakban a kaposvári színházban dolgozott koreográfusként.

— Valóban, de igen ellentmondásos volt a légkör. Először támogattak. Fejlődtem is, de hamarosan visszahúzott a színház vezetésének beállítottsága.

A koreográfia megvalósul

— 1974-ben kivándorolt Svédországba.

— Igen, újrakezdeni az ismeretlenben. Filozófiával foglalkoztam és a tánc strukturalista elméletével. Információelméleti könyveket olvastam, és megfogott a kibernetika. 1975-től a svédországi táncfőiskolára jártam, és csak a tanulásnak éltem. Az újromanticizmus technikaellenessége azonban nem kezdvetett törekvéseimnek. Én úgy próbáltam „komponálni", mint egy zeneszerző. Moderntánc-együttest alakítottam Multimedia néven. Ekkor már számítógépen szerzett zenére készítettem koreográfiáit.

— Milyen különbséget tesz a hagyományos és a gépi zene között?

— Nem a zenében levő különbségek érdekelnek! A kommunikáció a forradalmi a rendszerben, az egzakt kölcsönhatás a tánc és a zene között.

— A számítógépre a programokat saját maga készíti el?

— Nagyrészt igen, turbo Pascalban. Most már bedolgoznak a Királyi Műszaki Főiskola tanulói is. Például az animációs program jó részét ők készítették.

— Mit szólt a szakma az új „komponálási" módszerhez és az egész számítástechnikai rendszerhez?

— A művészek furcsállták, de hagyták. Ellentmondásos érzés, hogy olyan közegben mozgok, ahol nem tudnak vitatkozni velem.

— Kutatónak vagy művésznek érzi magát?

— Elsősorban művésznek, koreográfusnak. A számítástechnikát a mozgás elemzésére, a zene és a tánc szerkezetének megfeleltetésére használom elsősorban, természetesen a notáción kívül. Egyébként a rendszert még nem tekintjük teljesnek; nagyon sok tisztázatlan részlet van.

Sok sikert kívánunk a fejlesztéshez!

PINKE GYÖRGY

DIGITAL szelvény Mikroszámítógép Magazin 1988. március

Egy sarokkal olcsóbb!!!

A DIGITAL Számítástechnikai Szaküzlet (1026 Bp. Szilágyi Erzsébet fasor 35.) Sinclair-termékekre szakosodott: elsősorban a ZX81-es és Spectrum gépekhez használatos eszközöket, programokat árul. De kínál más gépekhez való tartozékokat (például botkormányokat), számítástechnikai alkatrészeket (integrált áramköröket stb.) és zsebszámológépeket is.

Aki ebben az üzletben a lapunkból kivágott sarokszelvényt átadja, vagy megrendelésével együtt oda elküldi, minden hónapban más-más cikket olcsóbban vásárolhat meg. A kedvezmény a szelvényen feltüntetett hónapban érvényes. Minden árengedménnyel vásárolt darabhoz le kell adni egy szelvényt.

A bolt utánvéttel szállítást is vállal, és a szokásos 6 hónap helyett *1 év garanciát ad.*

Az e havi kedvezmény

Két botkormányos illesztő fényceruzával 2500,— Ft, engedmény 20%

Pisztoly formájú botkormány 780,— Ft, engedmény 17%

Két botkormányos illesztő fényceruzával

A Spectrum gépek buszcsatlakozójára köthető, kapcsolós (Atari, Commodore típusú) botkormányok illesztésére szolgál. A játékprogramokhoz az ún. Kempstone típus a szabvány, amely egyaránt alkalmas az egy és két botkormányos játékokhoz. Sok játékot billentyűzetről is lehet játszani, de mivel ez hamar tönkreteszi a billentyűket, célszerűbb botkormányt használni.

A külön illesztővel csatlakoztatott fényceruzával rajzolhatunk és törölhetünk a képernyőn, majd a kész ábrát eltárolhatjuk, magnetofonszalagra vihetjük.

Pisztoly formájú botkormány

A leggyorsabb botkormány. A benne lévő, találmány tárgyát képező szerkezetnek köszönhetően ez az egyetlen olyan típus, amelynél nincs szükség a törés megakadályozását szolgáló fékre. Ennek következtében olyan gyorssá vált, hogy valamennyi versenyben gyorsabbnak bizonyult a többi botkormánynál. Egy összehasonlító mérés szerint például a Quickshot II-höz képest 2,2-szeres a sebessége. Ebben a kísérletben 4 személynek a lehető legnagyobb sebességgel 100—100 jobbra-balra kapcsolást kellett végeznie mind a kétféle botkormánnyal. (A Commodore újság felkérésére is végeztek összehasonlító méréseket az itthon elérhető botkormányokkal — köztük ezzel is —, de az eredményről lapzártáig nem értesültünk.)

A fékhiány miatt igen kis erő, egy kisgyerek ujjának pöccintése is elég ennek a botkormánynak a működtetéséhez. Így lehetővé válik a nagyon pontos beállítás, ami például a rajzkészítő programoknál nagy előny. A bot — annak ellenére, hogy minimális erőhatásra is reagál — az erős rángatást is kibírja.

A konstrukció előnye még, hogy meghibásodás esetén egyszerűen javítható. A formája pedig a lövöldözős játékokhoz „testhezálló".

A pisztoly formájú botkormány a II. Mikroszámítógépes Találkozó hardverpályázatán a legötletesebb megoldás díját kapta.

S. E.

Az NJSZT kitüntetettjei

A hagyományokat folytatva, a Neumann János Számítógéptudományi Társaság 1987-ben is kitüntette élenjáró aktivistáit. Az Országos Elnökség határozata alapján az alábbi tagok részesültek kitüntetésben:

Neumann János-díj

Kertész Ádám
Tóth Istvánné
NJSZT, ügyvezető főtitkárhelyettes
Pál László
OMFB, főcsoportfőnök

Kalmár László-díj

Arató Mátyás
SZÁMALK, tudományos tanácsadó
Cser László
BME, számítóközpont-vezető
Csaba László
MTA—SZTAKI, főosztályvezető

MTESZ emlékplakett

Dr. Kovács Péter
MNB, főosztályvezető-helyettes

A számítástechnika eredményes oktatásáért és a számítástechnikai kultúra terjesztéséért Tarján Rezső-díjat kaptak:

Dusza Árpád
országos számítástechnikai szaktanácsadó
Kovács Mihály
ny. Piarista gimnáziumi tanár
Szlávi Péter
ELTE, programozó matematikus
Zsakó László
ELTE, programozó matematikus

Tanulmányi munkájukért kitüntetett egyetemi hallgatók

BME TDK-díj

Mandzsu Zoltán, elméleti villamosságtan tsz. a „Memóriaorientált digitális transzmultiplexer szimulációja" és

Bányai Ervin, folyamatszabályozási tsz. a „Tervezési módszer párhuzamos végrehajtású, több automatát leíró vezérlési folyamatok tervezésére és rögzített (szinkron fázisregiszteres) struktúrában történő megvalósításra" c. dolgozatával elért első helyért.

Az NJSZT diplomaterv- és szakdolgozat-pályázatán 1987-ben 11-en indultak. A bírálóbizottság úgy határozott, hogy a kiírtnál kevesebb díjat ad ki, tekintettel a kevesebb beérkezett pályaműre. Két pályázó kapott első és kettő harmadik díjat.

Első díj

Görbics Lívia, volt matematika—kémia—számítástechnika szakos hallgató „Számítógéppel csatolt kémiai demonstrációs kísérletek" c. szakdolgozatáért, melyet az ELTE TTK fizikai—kémiai és radiológiai tanszékén készített.

Paizerné Tóth Tímea, volt német—történelem szakos hallgató a „Computergestützter Deutschunterricht" c. szakdolgozatáért, melyet az ELTE BTK Audiovizuális Technikai Központjában készített.

Harmadik díj

Mohay Tamás, volt műszer- és irányítástechnika szakos hallgató a „Grafikus nyelv vezérlési feladatok megoldására" c. diplomatervéért, amelyet a BME VMK műszer- és méréstechnika tanszékén készített.

A legjobb főiskolai dolgozat szerzője

Wéber László, volt számítástechnika szakos hallgató „Vonalkódolvasó és -generáló program IBM PC/XT számítógépre" c. szakdolgozatáért, amelyet a Kandó Kálmán VMF Számítástechnikai és Matematikai Intézetében készített.

A BME Villamosmérnöki karán meghirdetett programozási versenyen

Verhás Péter műszaki—fizika B szakos hallgató,

BME „Kozma László" digitális technika versenyen elért első helyért

Pintér Gábor részesült elismerésben. Gratulálunk!

Hogyan válhatunk „kód és dekód fiai"-vá?

A számítógép-élvezet akkor ér a csúcspontjára, amikor az ember a tiszta hardverrel kerül szembe. Erre kétféle minőségben lehet módja: hardveresként és szoftveresként.

A hardveres esete persze kézenfekvő, hiszen a szakmájából következik, hogy hardverrel dolgozik, igaz, egyre több szoftver beütéssel. El kell ismerni, hogy a számítógépek igazi lehetőségei nem annyira a billentyűzethez, mint inkább a gép hátoldalán található interfészekhez kapcsolódnak. Most azonban nem a hardveresekkel foglalkozunk.

hogy gépeink nem ismerték a magyar ékezetes betűket. Erre persze a dokumentáció is utal, de a következményekről nem beszél. Nos, egyes kérdések megválaszolásánál kerülhetünk bajba.

A programok többnyire jól működnek, és valamelyest a dokumentáció is segít. Kár, hogy fogalomtárának néhány magyarázata hiányos. A tanulás interaktív módon, kérdések és válaszok formájában történhet. A hibás válaszokra a hibajelzés mellett érdekes informatív példák jelennek meg.

Terméknév:	Gépi kódú oktatóprogram-csomag
Forgalmazó:	Novotrade - Octasoft
Szerzők:	Bereznay Tibor, Pajor Gábor
Géptípus:	C16, C Plus/4
Hordozó:	három kazetta
Dokumentáció:	22 oldal
Ár:	606,- Ft

1. táblázat

Amikor az ember szoftveresként kerül szembe a hardverrel, igen kemény feladatot kap: mindent saját magának kell megszerveznie az üzembiztos működéshez. Aki erre vállalkozik, az a szoftveresek megkülönböztetett kasztjához tartozónak vallhatja magát: rangja minimálisan „gépi kódú" vagy „assembler" programozó, sőt „rendszerprogramozó", esetleg „rendszerszoftveres" — a szakzsargon szerint: „bitfaragó".

Egy hétköznapi szoftveresnek jószerivel fogalma sincs arról, hogy mennyire távol dolgozik a hardvertől, és hogy milyen kényelmi eszközökkel halmozza el már egy vacak beégetett BASIC interpreter is, nem beszélve az operációs rendszerekről. Ezen csak akkor kezd el gondolkozni, amikor a gépkönyv vagy egy jó cikk hatására felbuzdul, és a gépi kód vizenyős talajára lép. Rendszerint pillanatokon belül olyan vereségeket szenved, hogy örökre elmegy a kedve az ilyen kalandoktól. Csodálatosnak hitt gépe elkeseritően elbutul, sokszor még egy nyamvadt hibajelzést se tud kinyögni, vagy összevissza zagyvál.

Nos, ifjak — bátrak és merészek, tizévestől akárhány évesig —, vágjatok neki e nagyon nehéz mesterségnek, mert bár elsajátítása szívósságot igényel, azért nem reménytelen! Már csak azért sem, mert a jó kézikönyveken túl egyre több számítógépes oktatóanyag áll rendelkezésre. Ezek közül vizsgáltunk meg egyet, a Novotrade-Octasoft termékét, amely az iskolákban elterjedt Commodore 16 és Plus/4 számítógépeken használható. A programcsomag főbb jellemzőit az *1. táblázatban* foglaltuk össze.

A próbát C16 és C Plus/4 gépeken végeztük. Mint ilyenkor lenni szokott, most is bajok voltak a kazetták beolvasásával. A Novotrade azonban kölcsönadta Headjust nevű magnófej-beállítóját, és a fejbeállítás után tényleg be tudtuk olvasni a szalagokat. A beállítást egyébként nagyon gondosan kell végezni, különben a kazettán található turbó betöltő nem tud jól dolgozni!

Munkánkat egyáltalán nem könnyítette,

Sajnos be kell számolnunk néhány zavaró értelmezésről, sőt kifejezett hibáról is. A számrendszerekről szóló részben a program a számrendszer nagyságáról beszél a számrendszer alapszáma helyett. Az alapszámot ugyanis a számrendszerben használható jegyek számával azonosítja, ami félrevezető. Az ifjabb korosztály számrendszerekről szerzett ismereteit ez talán nem rendíti meg, de a 25 éven túliak között is lehetnek bátor vállalkozók, akik szeretnének gépi kódot tanulni!

A számítógépek a komplemens kód miatt a nullát (0) pozitív számnak tekintik. Az iskolai matematikaoktatás ezzel szemben a nem létező mennyiséget hangsúlyozza a nulla kapcsán, így annak előjele nagyon zavaró az iskolások számára. Erre több figyelmet kellett volna fordítani a magyarázatban.

Baj van a helyiérték, a valódi érték és a számjegyek értelmezése körül is. Nem a számok, hanem a számjegyek helyiértéke adja a valódi értéket. Nem a bináris számrendszer helyiértékei lehetnek 0 és 1 értékűek, hanem a számjegyei választhatók a 0 és 1 készletből. Továbbá angolul a kettes számrendszer jegyeit hívják Binary Digit(ből rövidítve bit)-nek, nem pedig a „számrendszer egyes állapotát".

Az alapfogalmakon túl a gépi utasítások, címzések tárgyalásában is találtunk rossz értelmezéseket és hibákat.

Az implied (elhagyható) címzésmódnál csak az angol szó szerepel, az immediate (közvetlen) címzésmódnál pedig csak a magyar megfelelővel találkozhatunk.

Az implied címzés magyarázata hiányos, ezért megpróbáljuk pótolni. Ez a címzésmód akkor használható, ha az operandus címe a műveleti kódból következik, ezért nincs szükség címrészbájtokra. Az, hogy az implied címzésmód csak regiszterekkel kapcsolatos, nem felel meg a valóságnak. A vermet használó utasítások ugyanis a főtárat is használják.

A tanuló elég nehezen értheti meg a relatív címzési módot, mivel annak definíciója sehol sem található meg a programcsomagban, sőt a dokumentációban sem. A relatív cím egy komplemens kódban ábrázolt előjeles érték, amely egy bázis(regiszter) értékhez adódik hozzá, hogy megkapjuk az operandus címét.

A címzésmódokat oktató program a programcsomag legalacsonyabb színvonalú része.

A TXA utasítás definíciójában az „áthelyezés XR—AC között" félreérthető, túlzottan az angol rövidítés lefordítására koncentrál. Jobb, ha azt mondjuk: a TXA az XR regisztert az AC-be másolja.

Az STA utasítás definíciója hibás. Helyesen az STA az AC tartalmát az operandusként megadott címre másolja.

Az egyik példaprogram a 7501 gépi kódjában nem létező TXY utasítás használata miatt működésképtelen. A feladat működő megoldása: LDX #$00; LDY #$00; DEY; BNE $FD; DEX; BNE $FA.

A program egyik szövege ez: „Írj 1-1 utasítást!" Eltartott egy darabig, amíg rájöttünk, hogy mit akar. Elsőre úgy értelmeztük, hogy gépi kódú utasítást vár. Később derült ki, hogy utasításmnemonikot szeretne kapni. Ráadásul a program szokásától eltérően nem reagált a hibás válaszra, így nem árulta el a megoldást. Egyébként nagyon korrekt a válaszok elbírálásában. A sűrűn előforduló érdekes keresztkérdésekre adott hibás válaszok esetén megjelenik a helyes válasz és a kimeritő magyarázat.

A program jól kihasználja a számítógép nyújtotta előnyöket: látványos szemléltető rajzocskákkal, filmszerű diagramokkal igyekszik megmagyarázni a bonyolult eseteket.

A programcsomaghoz tartozik egy egyszerű assembler is, de inkább „igazi" assembler használatát javasoljuk. A magát az assemblert megvalósító programot azonban érdemes tanulmányozni.

Hiányként róható fel a programcsomagnak, hogy nem tartalmaz olyan teszteket, amelyekkel a tanuló felmérhetné tudását. Aki már tud valamit, annak nem kellene végigvennie az összes leckét, hanem a számára homályos részekkel foglalkozhatna. Egy interaktív csomagtól ez joggal elvárható.

A hibák ellenére a programcsomag összességében „jó" minősítést kapott tőlünk. A minősítés részletezését lásd a *2. táblázatban.*

Végül megjegyezzük, hogy a programcsomag nem pótolja az egyébként bőséges választékban rendelkezésre álló, jó minőségű kézikönyveket.

ZSADÁNYI PÁL—ifj. ZSADÁNYI PÁL

2. táblázat

Kezelhetőség:	jó
Teljesség:	jó
Dokumentáltság:	közepes
Használhatóság:	jó
Ár/teljesítmény:	közepes
Összbenyomás:	jó

TVC

Univerzális perifériakártya

Ahogy a Magazin 1987/12. számában olvastuk, már több mint 5000 Videoton számítógépet adtak el az országban. Sajnos a gépekkel kapcsolatos hardver és szoftver ismeretek „eladását" nem szorgalmazza a Videoton, ezért az ilyen információk sokakat érdekelhetnek. Ezúttal csupán egy kis részt tárgyalunk: azt, hogy hogyan lehet a gép négy csatlakozósávjába helyezhető perifériakártya illesztését elvégezni. A gép kezelési útmutatója tartalmazza a négy 2×43 pólusú csatlakozó jeleinek kiosztását, de sajnos magyarázatok nélkül.

A perifériák illesztéséhez egyrészt az adatvezetékekre, másrész a címkiválasztó hálózatra van szükség. A TVC-ben ezt igen ötletesen úgy oldották meg, hogy a címet tartalmazó nyolc vezetékből a felső négyet belsőleg dekódolták, és ebből négyet, az 1XH, 2XH, 3XH, 4XH címeken megszólalókat (H hexadecimális alakot jelöl) kivezették a csatlakozósávra (A39 pontok, jelük: SLOT 0—3). A gépet a billentyűzet felől nézve az első 1XH kártyacím, a SLOT 0 a jobb oldali legszélső kártya, és utána sorban következik a többi.

Az alsó négy címvezeték 16-féle kombinációjával (X=0H . . . FH) tehát összesen 16 perifériacímet hozhatunk létre egy kártyahelyen belül.

Az elmondottak illusztrálására a 8 digitális be- és 8 digitális kimenetet tartalmazó perifériakártya elvi kapcsolási rajzát közöljük az *1. ábrán.*

A rajzon szereplő téglalappal bekeretezett kivezetések jelnevei megegyeznek a csatlakozókiosztásnál leírtakkal (lásd a kezelési útmutatót), így az adott kártyakivezetési pont onnan azonosítható.

Az U1 jelű VAGY kapukból kialakított hálózat az U2 áramkörrel együtt végzi a teljes címdekódolást. Az NBRD és NBWR jelek felhasználásával — az U2 felső négy kimenete — a bemeneti kapuk kiválasztását végezhetjük el, az alsó négy segítségével a kimeneti tárolókba írhatunk. Az így dekódolt perifériacímek rendre K0H—K3H, ahol K=1,2,3,4, attól függően, hogy melyik csatlakozóba helyezzük a kártyát.

Az összehuzalozott kártyát az 1. csatlakozóba helyezve, a következő BASIC programmal ellenőrizhetjük a KI1—KI8 kimenetek működését:

```
10 FOR I=0 TO 255
20 OUT 16,I
30 NEXT I
40 GOTO 10
```

A program futásakor a kimeneteket oszcilloszkóppal nézve, mindig felezett frekvenciájú, szimmetrikus négyszögjeleket látunk.

A BE1—BE8 bemenetekre kapcsolt logikai jeleknek megfelelő decimális érték folyamatosan az alábbi programmal jeleníthető meg:

```
10 A=IN(16)
20 PRINT A
30 GOTO 10
```

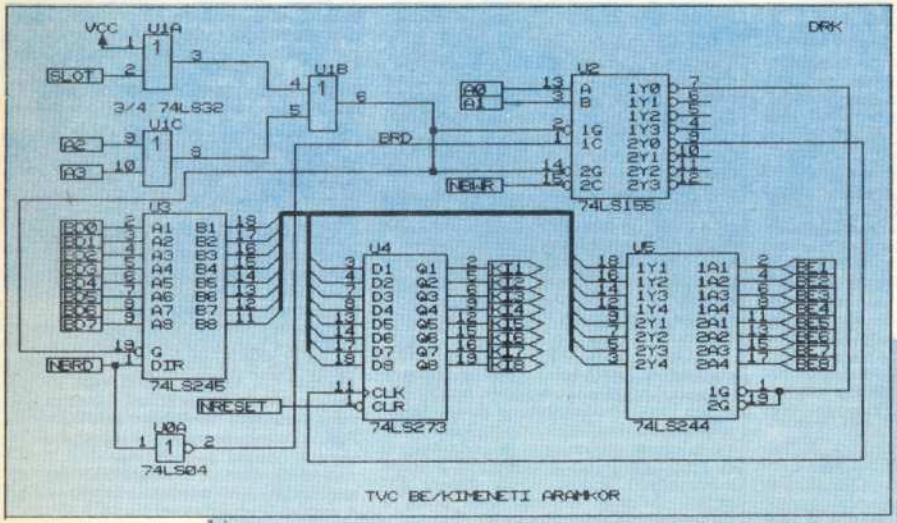
Az U4 és U5 jelű áramkörökből maximálisan 4-4 darab használható (azaz 32 be- és kimenet), amelyek vezérlésére értelemszerűen az U2 többi kimenetét használhatjuk fel.

Mivel minden bizonnyal nehézséget okozhat ennek vagy az ehhez hasonló kártyának az elkészítése, ezért a *2. ábrán* egy „csupalyuk" univerzális kártya NYÁK-rajzát is közöljük, amin az ilyen — és a későbbi számokban ismertetésre kerülő — kapcsolások könnyen összehuzalozhatók.

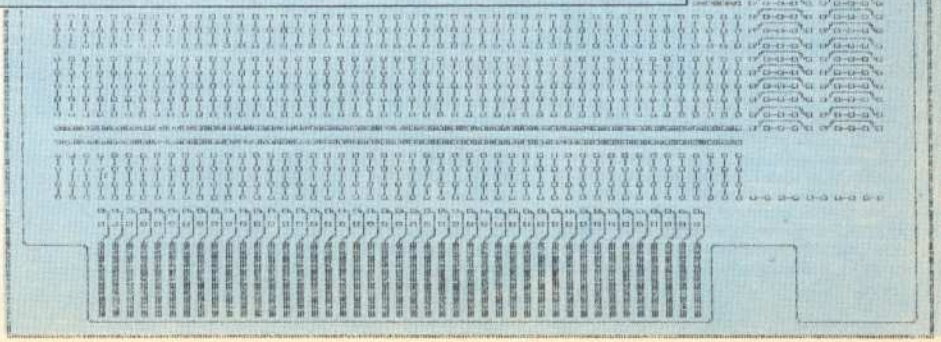
Célszerű az integrált áramköröket foglalatba helyezni és úgy behuzalozni, mert egy tokcsere sokkal egyszerűbben elvégezhető.

A szerkesztőségen keresztül ez az univerzális kártya 740 forintos áron megrendelhető.

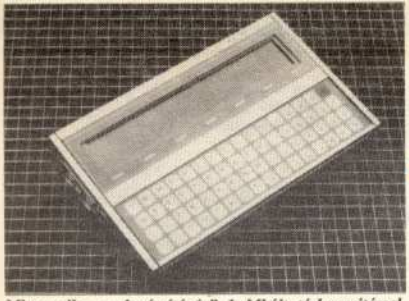
DR. KÓNYA LÁSZLÓ



1. ábra. TVC perifériakártya kapcsolási rajza



2. ábra. TVC univerzális kártya NYÁK-terve



Microscribe „zsebszámítógép" 1 Mbájt tárkapacitással és csatlakozási lehetőséggel

Mi és a számítógép — és a televízió, a video és egyebek

Mindenekelőtt köszöntöm az olvasókat, akik remélhetőleg nézők is, vagy azok lesznek. Mármint a Mi és a számítógép című műsor nézői a televízió képernyőjén. Azé a műsoré, amelynek szerkesztője vagyok.

Nemrégiben a két szerkesztőség elhatározta, hogy szorosabban működik együtt, mint eddig. Tettük ezt abban a reményben, hogy kedves nézőink, olvasóink megelégedésére szolgái majd.

Előljáróban úgy illik, hogy elmondjam, mit kínálunk a Mikromagazin olvasóinak. Mivel mint minden lapnak, a Mikromagazinnak is megvan a kialakult szerkezete, elsősorban olyan határterületeket választottunk, amelyek eddig nem szerepeltek, de amelyeknek a laphoz is, a tévéműsorhoz is közük lehet. Ha röviden kellene fogalmazni, azt mondhatnánk, hogy mindegyik kapcsolatos azzal, amit a múlt század közepétől éternek neveztek a fizikusok, de ma inkább elektromágneses jelekként vagy hullámokként emlegetjük őket.

A Mi és a számítógép című műsorban több mint egy éve rendszeresen jelentkezik egy új rovat, a programsugárzás. Aki a Rádió- és Televízióújságból, más lapokból vagy a műsorból nem ismerné, annak elmondom a dolog lényegét. Közismert, hogy a házi számítógépek a programok és adatok rögzítésére kazetta-magnetofont használnak. Minden számítógépes amatőr jól ismeri azt a sípolásszerű, egyesek szerint a torokfájós kakaséra emlékeztető hangot . . . Kézenfekvő a gondolat, hogy ez a hang, ha kazettán rögzíthető, akkor rádión vagy a tévé hangcsatornáján is kisugározható. A felhasználóknak csak rögzíteniük kell az adásból, és máris rendelkezésükre áll egy újabb számítógépes program.

A műsorban 1986 decembere óta állandóan szereplő programsugárzás tapasztalatai rendkívül kedvezőek. A programok az adásból jól rögzíthetők. A programsugárzáshoz kapcsolódó rejtvényeink nyertese a rövidesen megtartandó sorsoláson átveheti a fődíjat, egy C64 számítógépet.

Közben az is bebizonyosodott, hogy a programsugárzás a szoftver-cserebere igen hatékony eszköze lehet. Lelkes amatőrök eddig is több programot küldtek be. Most a Mikromagazin olvasóit is arra kérem, ne kíméljék a szerkesztőséget, és küldjék el azokat a programjaikat, amelyeket közkinccsé kívánnak tenni. A programsugárzás megkíméli viszont majd a kedves felhasználókat a begépelés, hibakeresés gyötrelmeitől. Egyetlen kérésünk van csak: saját készítésű programokat küldjenek, ugyanis nem szeretnénk szerzői jogi vitákba bonyolódni.

A szerkesztőség címe:
Mi és a számítógép szerkesztősége
Magyar Televízió
1810 Budapest

Természetesen várjuk a programokkal és rögzítésükkel kapcsolatos észrevételeiket, javaslataikat is.

A másik határterület, amellyel a műsor az utóbbi időben rendszeresen foglalkozik, a számítógép és a videózás kapcsolata. Napjainkban egyre több számítógép-tulajdonos jut videóhoz is, és egyre több videózó kezd érdeklődni a számítógépek iránt. Ez a kölcsönös érdeklődés nemcsak abból adódik, hogy nagyon kézenfekvő a gondolat, hogy a számítógépet video-feliratozásra lehet használni, hanem abból is, hogy ha egy jól sikerült videoklip láttán kicsit utánanyomozunk, a technikai segédeszközök között szinte mindig felfedezhetjük a számítógépet. Ahogyan a műsorban is visszatérünk rá, itt a lapban is rendszeresen foglalkozunk ezzel a témakörrel.

Végül, de nem utolsósorban még egy ajánlat. Előfordul, hogy a műsor stábjának megadatik, hogy egy-egy külföldi kiállításon, konferencián kutathassa: merre is fejlődik a számítástechnika. A műsorban rendelkezésre álló néhány perc sokszor kevés ezeknek a tapasztalatoknak a közreadására. Ezért ezeket szívesen osztjuk meg a Mikromagazin olvasóival is. Legutóbb a londoni COMPEC '87 kiállításon nyílt alkalmunk rövid tájékozódásra. Igyekeztünk a témákat nem anynyira a szakma, mint inkább a felhasználók oldaláról megközelíteni.

A COMPEC show — ahogy a londoniak nevezik — fizikai méreteit tekintve nem vetekedhet a párizsi SICOB-bal vagy a hannoveri CeBit kiállítással, de mindenképpen a jelentősebbek közé tartozik. Jellegzetessége, hogy a komplett rendszerek bemutatására helyezik a hangsúlyt: alkatrészt vagy részegységet viszonylag keveset láthattunk. Hardver- és szoftvertermékek egyaránt szerepeltek, szinte minden gépkategóriában, a személyi számítógépektől a nagygépekig. A valóságos vagy vélt fejlődési tendenciákat abból a szemszögből próbáltuk csoportosítani, hogy milyen sorsot szánnak — legalábbis a gyártók szándéka szerint — nekünk, felhasználóknak.

Az első, ami szembetűnő volt, a sokfajta hordozható, mini és zseb méretű számítógép. A nem is olyan távoli jövőben úgy látszik, már egy lépést sem tehetünk számítógép nélkül. Ez olyannyira komolyan értendő, hogy ezek között az apróságok között például vízálló típus is található. Lebecsülni már csak azért sem szabad őket, mert egyesekben nem kevesebb, mint 1 Mbájt memória rejtőzik.

A kicsi — de nemcsak kicsi — számítógépek és főleg használóik lehetőségeit megsokszorozza, hogy a már meglévő vagy a most kiépülő telekommunikációs hálózatokon keresztül kapcsolatba tudnak lépni egymással és az egészen nagy géprendszerekkel is. Ezeknek a kommunikációs hálózatoknak a rohamos fejlődése a másik megfigyelhető tendencia. A számítógépes munkahelyeknek egységes kommunikációs hálózatba történő integrálásától a szellemi és az adminisztrációs munka jelentős felgyorsulását várhatjuk. Gondoljunk például arra az esetre, amikor egy Magyarországon készült tervet Törökországban vagy éppen Ausztráliában kell jóváhagyatni, vagy egy engedélyt beszerezni. A számítógépes kommunikációval elmaradhat az oda-vissza utazgatás, és az információcsere mégis tökéletesebb és gyorsabb lesz.

A computer szóból a com szótagot lassan kénytelenek leszünk átértelmezni. A különböző hálózatok fantázianeveiben ez a szótag ugyanis már nem a számítástechnikára, hanem a kommunikációra és egyre inkább a telekommunikációra utal.

6,6 Mbájt kapacitású hajlékonylemez-meghajtó

A számítógépek tulajdonosai, felhasználói rengeteg adat birtokába juthatnak. Ezek között az eligazodás nem is egyszerű feladat. A számítógép-alkalmazásnak már szinte külön ága az ún. üzleti grafika, amely segít áttekinteni például hogy miből, hol, mikor, mennyink van és mindez hogyan változik az idők során. A kiállításnak szinte nem volt olyan standja, ahol grafikonokat ne láttunk volna a gépek képernyőjén.

Ezzel már újabb alkalmazási területhez érkeztünk: a számítógépes grafikához, amely sajátos módon elég jó fokmérője egy-egy processzor, gép vagy rendszer képességeinek. A képek tárolása, megjelenítése, esetleg mozgatása rengeteg információ rögzítését és feldolgozását igényli. Érthető okból mi, televíziósok, igen szívesen használjuk ezt a mércét.

Ugyancsak jól megfigyelhető jelenség a számítógépek műveletvégzési sebességének gyors növekedése. Egymás után jelennek meg az egyre hosszabb adatsoron, egyre nagyobb gyorsasággal műveletet végezni tudó mikroprocesszorok. Napjainkban azonban valami egészen új megszületésének lehetünk tanúi. A mai számítógépek akármilyen gyorsan, de alapjában véve soros módszerrel dolgoznak. A processzor kivesz egy adatot a memóriából, csinál vele valamit, az eredményt visszateszi a memóriába, aztán rátér a következő műveletre. A COMPEC-en azonban új fogalom kerülhetett a számítástechnikai szótárunkba: „transputer". Erről John A. Marshall úr, a Gemini Computer Systems Limited vállalat menedzser igazgatója tájékoztatott bennünket.

„Vállalatunk az INMOS cég által gyártott transputer chipeket alkalmazza különböző számítógépeiben. A transputer processzorok működésének lényege, hogy magában a processzorban a műveletvégzés párhuzamosan történik. Több, sőt mondhatnám sok processzor dolgozik az áramkörben, ezek mindegyikéhez tartozik egy kis darab memória, amely szintén az áramkörben található. Minden processzor igen gyorsan tud dolgozni a saját memóriájával, és igen gyorsan tud kapcsolatba lépni a többi processzorokkal. Az eredmény hihetetlenül gyors műveletvégzés. Többen azt jósolják, hogy rövidesen a Cray képességeivel rendelkező számítógépek kerülhetnek a dolgozóasztalunkra. (A Cray ma egy sokmillió dolláros szuperszámítógép.) Mások az ember—gép kapcsolat újabb forradalmát várják a transputer chipektől. Cégünk jelenleg a meglévő számítógép-konstrukciók működésének gyorsítására épít be transputereket. Ezeket alkalmaztuk képfeldolgozó rendszerünkben."

Mi tagadás, a látvány lenyűgöző volt. Hat képsikon video minőségű kép, folyamatos animációval. Mindezt természetesen az asztali méretű, legújabb Gemini Challenger típus produkálta.

Mielőtt azonban bárki gyorsan túladna meglévő masináján, megemlítek két dolgot:

— a tömeggyártásig és a magyarországi forgalmazásig biztosan eltelik még egy kis idő,

— úgy tűnik, a tervezők a régi felhasználókra is gondoltak. Bemutatták az IBM PC kompatibilis transputer kártyát, és láthattuk a Nyugat-Európában már házi számítógépnek minősülő Atari ST transputerizált változatát, amint éppen egy Hanoi-torony játékot rakosgatott szédítő sebességgel.

HEGYI ISTVÁN

Szekvenciális fájlt kezelő rutin

Számos feladat megoldásához célszerű szekvenciális fájlokat használni. Ezek kényelmes kezeléséhez készítettem az alábbi rutint, amellyel tetszőleges nevű fájlt lehet létrehozni, olvasni, bővíteni, törölni.

A főprogram és az 5000-től kezdődő szubrutin között az információcsere négy változó, az F$, MT$, NF és MD segítségével történik. A szubrutin számára az F$ és MD bemeneti, a másik kettő be/kimeneti változó. A bemeneti változók értékeit a főprogramban, a szubrutin meghívását megelőzően kell beállítani. Az F$ változóba kerül a fájl neve; a programlistán látható főprogramban — az első hat sorban — ez éppen ZIZI. Az MT$ munkatömb szolgál a fájl elhelyezésére. Egy tömbelem egyenlő egy rekorddal. A tömb méretét a főprogramban kell meghatározni: a kezelendő legnagyobb rekordszámú fájl rekordszámával egyenlő. Esetünkben ez éppen 200, a 10-es sor szerint. Mivel bármilyen fájl átmeneti tárolására ez a tömb használatos, szükségünk van egy újabb változóra, az NF-re, amely az utolsó tömbelem indexét tartalmazza.

Írás és bővítés esetén az NF értékét szubrutinhívás előtt meg kell adni, hiszen az MT$ tömb elemei az MT$(o)-val kezdődően és az MT$(NF)-fel bezárólag kerülnek ki a fájlba. Olvasáskor az NF az utolsó beolvasott rekord helyét adja meg. Végül az MD változó a kívánt tevékenység (olvasás, írás, bővítés, törlés) beállítására szolgál.

A közölt példában, a főprogramban tehát a ZIZI nevű fájl beolvasása, majd képernyőre listázása történik. Fájl törléséhez elég a nevet és MD=3-at megadni.

DR. SZIKSZAI CSABA

```
10 DIM MT$(200)
20 F$="ZIZI"
30 MD=0
40 GOSUB 5150
50 FOR I=0 TO NF:PRINT MT$(I):NEXT
60 END
5000 REM **********************
5010 REM * SZEKVENCIALIS FILE
5020 REM *       KEZELES
5030 REM **********************
5040 REM VALTOZOK:
5050 REM              F$ - FILENEV
5060 REM DIM MT$(200) - MUNKATOMB
5070 REM NF - A MUNKATOMB ERV.HOSSZA
5080 REM EN,EM$,ET,ES - HIBAVALTOZOK
5090 REM              ST - ALLAPOTJELZO
5100 REM              MD - TEVEKENYSEGJELZO
5110 REM                   0 = OLVASAS
5120 REM                   1 = IRAS
5130 REM                   2 = BOVITES
5140 REM                   3 = TORLES
5150 IF MD<0 OR MD>3 THEN RETURN
5160 OPEN 15,8,15
5170 ON MD GOTO 5370,5470,5570
5180 REM OLVASAS
5190 OPEN 2,8,2,F$+",S,R"
5200 GOSUB 5610
5210 NF=0
5220 S$=""
5230 GET#2,X$
5240 IF ST=64 THEN 5320
5250 IF ST<>0 THEN GOSUB 5610
5260 IF X$=CHR$(13) THEN 5290
5270 S$=S$+X$
5280 GOTO 5230
5290 MT$(NF)=S$
5300 NF=NF+1
5310 GOTO 5220
5320 MT$(NF)=S$
5330 CLOSE 15,8,15
5340 CLOSE2
5350 RETURN
5360 REM IRAS
5370 OPEN 2,8,2,F$+",S,W"
5380 GOSUB 5610
5390 FOR I=0 TO NF
5400 PRINT#2,MT$(I)
5410 IF ST<>0 THEN GOSUB 5610
5420 NEXT
5430 CLOSE 15,8,15
5440 CLOSE2
5450 RETURN
5460 REM BOVITES
5470 OPEN 2,8,2,F$+",S,A"
5480 GOSUB 5610
5490 FOR I=0 TO NF
5500 PRINT#2,MT$(I)
5510 IF ST<>0 THEN GOSUB 5610
5520 NEXT
5530 CLOSE15,8,15
5540 CLOSE2
5550 RETURN
5560 REM TORLES
5570 PRINT#15,"S:"+F$
5580 CLOSE 15,8,15
5590 RETURN
5600 REM PARANCS-CSATORNA OLVASAS
5610 INPUT#15,EN,EM$,ET,ES
5620 IF EN=0 THEN RETURN
5630 PRINT"■■■■■■■■■■■HIBA A LEMEZEN!"
5640 PRINTEN;EM$;ET;ES
5650 CLOSE 15,8,15
5660 END
5670 REM LEMEZHIBA-KEZELES
5680 PRINT"■■■■■■LEMEZHIBA SZAMA: ";ST
5690 CLOSE15,8,15
5700 END
```

256 szín ZX–Spectrumon

Ezt a képtelennek gondolt dolgot, mint szivárványt feszít ki a zivataron áttűző nap, oly könnyedén valósítja meg az alábbi program. Nézzük meg, hogyan?

A Spectrum alapállapotban 8 színt ismer. Mihelyst ezeket összekeverjük, máris 64 színt csináltunk. Ha a színeket kétféleképpen keverjük össze, akkor még mindig csak 128 árnyalathoz jutottunk. De a Speki ismeri ám az extrafényességet is, ami által minden színnél duplázza az árnyalatokat. Máris kész a 256 szín! Hogy hogyan lehet a színeket keverni, erre mutat példát a program.

Mielőtt rátérnénk a program tanulmányozásához és használatához szánt közlésekre, vegyük szemügyre a gyengéit is. Egyes árnyalatoknál igen élesen kitűnik a keverés módja. Nem megfelelő beállítású tévén a színek „hullámzanak". Mind a háttér-, mind a tintaszínt is felhasználja, ezért egy karaktermezőben csak az adott keverékszín vagy annak elemei lehetnek.

Néhány tanács a program beírásához. A nyomtató nem ismeri sem a grafikus, sem az UDG karaktereket. Ezért a listához a következő magyarázat szükséges: a 9808-as sorban az idézőjelek között 32 darab grafikus 3-as.

9846 IF c$ = „grafikus A" . . . .
9847 . . . : LET c$ = „grafikus B"
9901 . . . AND c$ = „grafikus B"

A program olyan egyszerű, amilyen tökéletlen. Az első részben előállítja az UDG-t, majd kirajzolja a keverékszíneket. Megkérdezi, hogy lapozzon-e. Ha igen, újabb színeket rajzol ki. Ezután a kívánt színt teljes méretben reprodukálja, majd kiírja, milyen tinta- és papírértékek mellett valósult meg, valamint azt is, hogy melyik grafikus karaktert használta.

A programhoz eredményes időtöltést kíván e sorok írója:

TURAY GÁBOR

```
2 ,1;"PAPER ";A;" INK ";B;" GRAPHICS A"
9702 FOR N= 0 TO 7
9703 READ M:POKE USR "B"+N,M
9704 NEXT N
9705 DATA 170,85,170,85,170,85,170,85
9710 FOR N=0 TO 7
9720 READ M: POKE USR "A"+N,M
9730 NEXT N
9740 DATA 255,0,255,0,255,0,255,0
9750 POKE 23609,40:BORDER 0: PAPER 0
9751 INK 7:CLS
9805 PRINT "256 SZIN ZX-SPECTRUMRA!!!"
9808 PRINT "                                ";,,
9809 PRINT AT 2,12;"1. OLDAL":LET C$="A"
9810 GO SUB 9950
9820 PRINT A,B;">";:FOR C=0 TO 4
9821 PRINT PAPER A;INK B;C$:NEXT C
9830 IF Y=00 THEN GO TO 9841
9840 GO TO 9810
9841 PRINT "MOST MEGMUTATOK ONNEK 128 ";
9842 PRINT "SZINT NORMAL FENYESSEGBEN.";
9843 PRINT "EXTRAFENYESSEGGEL 256-OT ";
9844 PRINT "ERHET EL. ":LET L$=""
9846 IF C$="A" THEN INPUT "LAPOZAS (I/N)";L$
9847 IF L$="I" THEN PRINT AT 2,12;"2. "
9848 LET C$="B":GO TO 9810
9850 INPUT "MELYIKET REPRODUKALJAM?";A$
9860 IF LEN A$<>2 THEN GO TO 9850
9880 LET A=VAL A$(1):LET B=VAL A$(2):CLS
9890 FOR E=1 TO 704:PRINT PAPER A;INK B;C$:NEXT E
9901 IF A$(1)<>A$(2) AND C$="B" THEN PRINT AT 12,30;"B"
9910 IF A$(1)=A$(2) THEN PRINT AT 12,7;"AZ ON SZINE: "A$(1);"!"
9920 PRINT #0;"NYOMJON AKARMIT A FOLYTATASHOZ!":PAUSE 0
9930 GO TO 9805
9950 READ Y:LET A$=STR$ Y:IF Y<10 THEN LET A$="0"+A$
9960 IF Y=00 THEN RESTORE 9980
9970 LET A=VAL A$(1):LET B=VAL A$(2):RETURN
9980 DATA 22,30,32,14,34,36,26,37,04,15
9981 DATA 24,35,74,27,66,77,65,67,75,64
9982 DATA 71,61,45,25,60,05,44,21,31,23
9983 DATA 20,01,55,47,63,73,51,41,43,53
9984 DATA 52,42,50,13,40,03,02,11,62,72
9985 DATA 57,56,46,12,17,54,06,16,76,33
9986 DATA 10,07,70,00
9999 CLEAR :SAVE "SZINMIXER" LINE 9700
```

Integrált szofver

A táblázatkezelő program használata I.

Lapunk 1987/11. számában már ismertettük a táblázat felépítését, így erre most nem térünk ki. A táblázaton belül a program indításakor a kurzor mindig az AA mezőben van, és a négy kurzormozgató billentyűvel léptethetjük mezőről mezőre. Ha ki akarnánk lépni a táblázatból, a program hangjelzést ad. Ha csak a képernyő határain lépünk túl, a táblázat újrarajzolódik, és az újonnan belépő mező is teljes szélességben látszik.

Mindezt könnyen kipróbálhatjuk. Ha az AA mezőből felfelé vagy balra ki akarunk lépni, az eredmény egy „beep" hang lesz. Jobbra vagy lefelé viszont léptethető a kurzor. Figyeljük meg, hogy a képernyőn a táblázat fölötti sorban a kurzorpozíciót jelző két betű mindig a kurzor aktuális helyének megfelelő karaktereket mutatja — például bal felső sarok: AA. A > jel használatával az egyik mezőből a másikba ugorhatunk: például a ZZ pozícióra a >ZZ utasítással és az ENTER lenyomásával. Hasonlóan juthatunk vissza a kiinduló helyzetbe: >AA és ENTER.

Figyeljük meg az adatbeviteli sort, ahol a kérdőjel (?) van. Írjunk be egy számot, például 12-t. A pozíciójelző minden szám leütésével jobbra mozdul. Ha minden számjegyet leírtunk, nyomjuk le az ENTER gombot. Hatására a 12 megjelenik az AA pozíción, a táblázat feletti sorban pedig, amely a tartalmat jelzi, a következőt láthatjuk: [AA] (V) 12. Vagyis az AA pozícióra számértéket vittünk be, ami jelen esetben 12. Most mozditsuk lejjebb a pozíciójelzőt, az AB mezőre, és írjuk be: 22, majd ismét ENTER. Ezzel a módszerrel a táblázat különböző pozícióira írhatunk be konstansokat.

Egyszerű műveletek végzése

Mozditsuk lefelé a pozíciójelzőt például az AC mezőre, és adjuk össze az AA mező tartalmát az AB mező tartalmával: +AA+AB, majd az ENTER gomb lenyomására az AC pozíción megjelenik az összeg: 34. Az első + jelet azért kell beírni, hogy a program „észrevegye", hogy aritmetikai műveletet végzünk. A második + jel az összeadás jele.

Az aritmetikai műveletek: az összeadás (+), kivonás (−), szorzás (•), osztás (÷) és a hatványozás (^) egyaránt használhatók. Az említetteken kívül további két műveleti jelet alkalmazunk: a % és a # jelet. A százalékjel itt nem százalékszámítást, hanem egész számot jelent, hasonlóan a BASIC nyelv jelöléséhez. Ha egy számítási formulába valahol beírjuk a % jelet, az addig a pontig eltávolítja a tizedeseket. Tovább nincs hatása. A kettős kereszt (#) az abszolút érték számításának jelölésére szolgál.

Az eddig megismert lépéseket célszerű néhány egyszerű számítással kipróbálni. A műveleti utasítások egyaránt használhatók formulában leírva és konstansokkal való számításoknál.

Nézzük először a kalkulátor üzemmódot. Mondjuk valamilyen számításhoz tudnunk kell, hány perc van egy munkahétben. Nos, egy munkahét percekben 5 (nap) * 8 (óra)*60 (perc)=2400 perc. Ha a műveletek eredményét például az AE pozíción akarjuk tárolni, akkor a számítás megkezdése előtt a kurzort oda kell vinni, majd leírni a műveleteket. A kulkulációs számsor a legfelső sorban lévő kérdőjel mögött fog megjelenni a következő formában: ?5*8*60. Az utolsó szám beadása után nyomjuk le az ENTER gombot. Az eredmény megjelenik a második sorban, valamint az AE pozíción. A második sor ilyen alakú lesz: [AE] (V) 2400. Jelentése: AE pozícióra értéket (V) tárolunk, jelen esetben 2400-at.

Fontos tudni, hogy csak a számítás végeredménye kerül a tárolóba, a kiszámítás menete nem! A műveletek végrehajtása minden esetben balról jobbra történik, amire néhány számításnál esetleg figyelni kell.

Számítási formulák bevitele

Maradjunk még a kurzorral az AE mezőben, és helyezzük egy fomulával definiált számítás eredményét az előbb kiszámított érték (2400) helyére. Írjuk be: +AA−AB és kérjük le a formulát az ENTER gomb lenyomásával. A második sorban ezt látjuk: [AE] (F) +AA −AB. Az ENTER lenyomására az AE pozíción megjelenik az eredmény: −10. Emlékeztetőül AA=12, AB=22, ahogy az a képernyőn látható. Ha nem működik a formula, az a legvalószínűbb oka, hogy az elsőnek megadott pozícióazonosító (itt AA) elé elfelejtettük beírni a + jelet. Ilyenkor ugyanis a program címkének értelmezi a beírtakat, és annak hatására számítás nem történik.

Próbáljunk ki most néhány más műveletet. Tegyük például az AG mezőbe +AA*AB értékét, +AA/AB legyen az AI pozíción és +AA^AB kerüljön az AK pozícióra.

A kurzor mozgatásával kapcsolatban megjegyezzük, hogy bármit is írunk a képernyő első sorába a ? után, a pozíciójelző csak jobbra és lefelé mozgatható a kurzormozgató gombokkal. Ebben az esetben a kurzor mozgatása (átlépés egy másik mezőbe) azonos hatású az ENTER gomb lenyomásával. A balra lépést a BACKSPACE, a felfelé mozgatást az exponens jele teszi lehetővé. Jobb módszer, hogy a betűváltó gombot (SHIFT) nyomjuk le, mert ekkor minden kurzormozgató gomb normálisan működik, úgy, mintha az első sor (a prompt line) üres lenne.

Vegyük sorra a program utasításkészletét! Ahhoz, hogy ezt megfigyeljük, mozgassuk a pozíciójelzőt az egyik olyan mezőre, amely tartalmaz például egy formulát. Legyen ez az AK pozíció. Ekkor nyomjuk le a / jelet. A formula eltűnik erről a helyről, és a következőt látjuk a képernyő második sorában:

— T TT TP S B V SZ N U M E

Mielőtt ezekre az utasításokra rátérnénk, lássuk a szerkesztés üzemmódot. Nyomjuk le az S betűt, majd az ENTER-t. A formula megjelenik az első sorban. A visszaléptetés segítségével az esetleg módosítani kívánt betűre vagy jelre lépünk, és egyszerűen felírjuk. Lehetőség van arra is, hogy a formulát egy másik helyre átvigyük, mialatt az eredeti helyén is ott marad. Nyomjuk le a SHIFT gombot, és az egyik kurzor mozgatásával léptessük a pozíciójelzőt a kívánt helyre: az ENTER lenyomására a formulamásolat ott megjelenik.

Újraszámolás

Lépjünk vissza a kiinduló pozíciónkra (AA), és írjuk be az eredeti 12 helyére például 10-et, majd nyomjuk le az ENTER gombot. 10 bekerül 12 helyére, de a képernyő más helyein, ahol a számítási formulában szerepel az AA mező értéke, nem történik változás, így az értékek nem jók. Érdemes megjegyezni, hogy a gép kiszámítja a formulát, amikor beírjuk, de ezután bármi

változtatást végzünk az adatokon, a programot utasítani kell, hogy végezze el a teljes táblázat újraszámítását. Ezt annyiszor megtehetjük, ahányszor csak akarjuk, de a több mint 600 mező újraszámolása némi időt vesz igénybe. Az újraszámoláshoz nyomjuk le a felkiáltójelet (!): a képernyő jobb felső sarkában ez megjelenik. Ebből tudjuk, hogy a program elkezdte az újraszámolást, ami kb. 8 másodpercig tart; ehhez jön még az az idő, amit a formulák kiszámításával tölt a gép. Amint a számolás elkészült, a képernyőn az új értékeket látjuk, és az utasítássor (első sor) ismét aktív lesz.

Ezek után vegyük sorra a rendelkezésre álló utasításkészletet. A /E (eltárol) utasítással a táblázat tartalmát szalagra menthetjük. A program megkérdezi a fájl nevét (ami nem lehet 8 karakternél hosszabb, és betűvel kell kezdődnie), és az ENTER lenyomása után elkezdi a mentést. Ez elég hosszú ideig tart, mert 2028 (3*676) pozíciót kell a szalagra kimenteni. Amíg a képernyőn fenn van a „SAVING /fájlnév" üzenet, a program működik.

A már rögzített táblázat tartalmát a /V (visszatölt) utasítással lehet betölteni. A program itt is megkérdezi a fájl nevét, és az ENTER hatására betölti a táblázat tartalmát, majd hozzáteszi, összefésüli az éppen munkában levő táblázat tartalmával. Ha tehát kizárólag egy régebbi táblázattal akarunk dolgozni, akkor ezt egy üres táblára kell rátölteni!

A teljes táblázat tartalmának törlése az operatív memóriából a /T (törlés) utasítással végezhető el.

Sok esetben szükség lehet a táblázat részeinek törlésére is. Ezt a következő utasításokkal tehetjük meg:

/TT (táblázat törlése) Csak a beírt számértéket törli, a formulák és a címkék változatlanul maradnak.

/TO (töröl oszlopot) Törli a kijelölt oszlop tartalmát.

/TS (töröl sort) Törli a kijelölt sor tartalmát. Címkék úgy törölhetők, hogy szóközt írunk a helyükre. A formulát törli, ha valamilyen értéket írunk a helyére, beleértve a Ø-t is.

/N (nyomtatás) Használatakor a program megkérdezi, hogy honnan, melyik oszloptól induljon a nyomtatás (bal oldal) és meddig tartson (jobb oldal). Néhány esetben, amikor a táblázat nagyobb, mint a nyomtató szélessége, részletekben kell a nyomtatást elvégezni.

/SZ (számolás) Ezzel az utasítással a számolás rendjét adjuk meg. Ha a számolást oszloponként kérjük, akkor az utasítás /SZO lesz, a kiindulópont az AA pozíció; ha soronként, akkor az utasítás /SZS és a kiinduló pozíció itt is az AA.

@SUM Ezzel az utasítással oszloponként vagy soronként összegzést végezhetünk. Az összegzést oszloponként felülről lefelé hajtja végre a program, sorok esetén balról jobbra. Az összegzés irányát (sor vagy oszlop) az O, illetve az S kiterjesztés adja meg. A kiinduló pozíciót is meg kell adni. Így az utasítás végleges formája például ez lehet: @SUMOAA. Jelentése: összegzés az oszlopon, az AA pozícióból kiindulva.

/TP (tizedes pont) Azt adhatjuk meg vele, hogy hány tizedes legyen. A teljes formátum: /TPØ vagy /TP1 vagy /TP2 stb. Az alaphelyzet /TPØ, azaz egész számokkal dolgozunk.

/B (betoldás) Segítségével új oszlopokat (/BO) és új sorokat (/BS) szúrhatunk be a már elkészült táblázatba.

Figyelem! A beszúrás után a mezőkre történő hivatkozások nem korrigálódnak automatikusan!

Legközelebb néhány összetettebb példán mutatjuk be a táblázatkezelő működését.

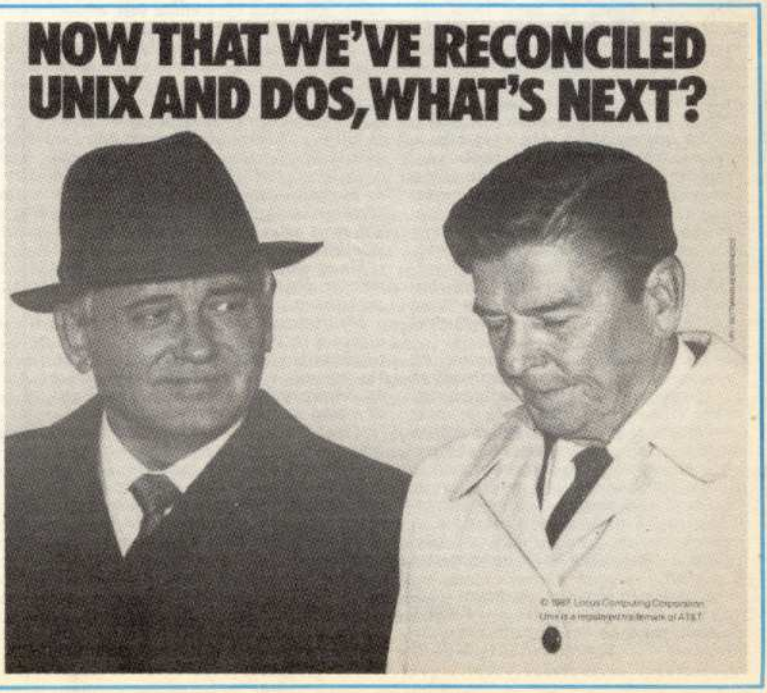
GALINA FERENC

UNIX–MSDOS találkozó

A Locus Computing Corp. cég az itt látható fényképekkel hirdeti PC-Interface nevű termékét. (A kép feletti szöveg: **Most, hogy mi összebékítettük a UNIX és DOS rendszert, mi a következő?**)

A cég terméke olcsó eszközzel teszi lehetővé, hogy az MSDOS-környezet megtartása mellett a felhasználó kihasználhassa a UNIX rendszer által biztosított többfelhasználós, időosztásos lehetőségeket.

S. E.



ADOM A MAGYARÁZATOT!

Egy C16 probléma

Visszatérünk egy régi kérdésünkre, melyre nagyon hosszú idő után kaptunk választ. A probléma többeket érint, a válasz pedig érdekes; reméljük, sokan szívesen olvassák.

Az 1987/2. szám 20. oldalán jelent meg ifj. Monfera Róbert kérdése a C16-tal kapcsolatban. Problémája az volt, hogy kb. 8 kbájtnyi BASIC program újraszámoztatása során a gép furcsa dolgokat művelt. Én feltételezem, hogy semmilyen gépi kódú programot, illetve rutint nem tartalmazott a BASIC program, legalábbis annak a lehetősége nem állt fenn, hogy a rutin a BASIC-területre írjon, továbbá kizártnak veszem, hogy a számítógép meghibásodott. Ebben az esetben porgramozói tévedés történt.

Szerintem a kérdező a program írása közben hibát vétett, amit nem vett észre. A hiba: egy olyan GOTO, GOSUB vagy RESTORE utasítás, amely nem létező sorra mutatott. Ezután futtatás nélkül újraszámoztatta a programot. Az eljárás során nem derült ki, hogy rossz egy vagy több hivatkozás, az interpreter ilyen esetben 65535-ös sorszámot generál. De másodszorra számoztatva (új parancs!), újra találkozott ezzel a sorszámmal. Mivel ez már fatális hiba, SYNTAX ERROR üzenettel az átszámozást megszakította. Ha ez a hiba a program közepén, esetleg a végén van, az előtte lévő sorszámokat már átírta. Innen ered az elszámozás. Ezzel együtt létrejöhet hiba magában a tárolt programban is (de most nem akarok a gép interpreterjének működésébe és a program tárolásának miképjébe belemélyedni). Ez magyarázatot ad arra, hogyan lehet egy újrasorszámozás után akár az egész progamot (látszólag!) elveszíteni.

Az a megoldás, amit a kérdező írt (mégpedig, hogy monitor üzemmódban a regiszter értékét megváltoztatta, így nyerte vissza a programot, még ha rossz sorszámokkal is), csak véletlen. Hiába írja át bármely regiszter értékét, az ténylegesen nem változik, a monitor csak a $0554—$0558-ig terjedő memóriában tárolja el. Ténylegesen csak program indításakor (G parancs) töltődnek be az értékek. Viszont a monitorból való kilépéskor a gép csak BASIC melegstartot hajt végre.

Az igazán jó módszer szerintem a következő. Meg kell nyomni a RESET gombot, majd be kell gépelni ezt a parancssort: POKE 4097, 1:RENUMBER. Ekkor természetesen semmiféle átszámozás nem történik, viszont lefut az interpreternek az a része, amelyik a BASIC sorokat kapcsolja (linkeli) össze. Ez a módszer mindig működőképes, sőt egyenlő egy RENEW paranccsal. A csak linkelés: SYS 34840. Ismétlem: mindez csak akkor igaz, ha gépi kódú rutin nem írt a BASIC-területre!

Végül annyit, hogy az egyik program, a játékturmix közel 24 kbájtos, és ezt is hibátlanul számozta újra a gép!

Még egy javaslat. Ha valaki nagyobb programot számoztatott újra, érdemes ellenőriznie, hogy volt-e hiba. Az ellenőrzés menete a következő. Meg kell hívni a monitort, majd a $2D—$2E memóriacímen meg kell nézni a BASIC program végét. Tételezzük fel, hogy itt azt találjuk: 12 30 . . ., ami azt jelenti, hogy a BASIC program vége $3012-n van. Ezután azt kell megkerestetni, hogy hol van a következő bájtsorozat: 36 35 35 33 35 a H 1000 3012 36 35 35 35 parancscsal. Ugyanis ha ezt találja a gép, akkor ott 65535-ös szám, vagy ami valószínűbb, egy olyan hivatkozás van, amivel eredetileg az interpreter nem tudott mit kezdeni (nem talált olyan sort), ezért oda egy ilyen sorszámot írt. Ha a keresés sikertelen, tehát egyetlen címet sem ír ki a gép, akkor az átszámozás hiba nélkül megtörtént. Ha talál olyan címet, és kiírja, az hibára utal. A megtalált memóriacím nagysága támpontul szolgálhat a hiba helyének felderítésére. Ebben az esetben — a program alapos ismeretével — könnyedén kijavíthatjuk a hibát, mert maga az átszámozás e hiba kivételével rendben megtörtént.

Ha az előbbi számsor nem vezet eredményre, érdemes az FF FF sorozattal is próbálkozni.

Megjegyzem, hogy a hiba okozója nemcsak a programozó, hanem az interpreter sajátságos működése is lehet. Ha ugyanis a RENUMBER parancsot ún. kétmenetes formában hajtaná végre, ez a hiba nem következhetne be. Mit jelent ez? Az első menetben táblázatban összegyűjti az összes ugráscímet az elugrási helyekkel együtt, anélkül, hogy a programban a címeket átszámozná, és ellenőrzi, hogy az ugráscímek létező sorra vonatkoznak-e vagy sem. Ha ilyenkor hibát talál, azt kiírja, és abbahagyja az átszámozást. Így nem hoz létre semmiféle hibás programot. A második menetben először átszámozza a táblázatban az ugrási címeket, majd ezután átírja a megfelelő sorokat.

KÁDÁR SÁNDOR

KI AD MAGYARÁZATOT?

A problémám az, hogy vannak olyan információk, melyeket eddig sajnos sehonnan sem tudtam megszerezni. Ehhez kérnék önöktől segítséget.

Kérdéseim a következők:

1. Mikroprocesszor
   AEC, RDY, IRQ, GATE IN, φOIN, P0—P7 kivezetései mi célt szolgálnak.
2. TED chip
   — főbb funkciói?
   — AEC, BA, CS0, CS1, COLOR, SYNC/LUM, SND, K0—K7 kivezetései mire jók?
3. 7700—10 chip (U16)
   — mi ez?
   — mi a feladata?
4. 6529 B chip (U13)
   — mi ez?
   — mi a feladata?

Gődér György

Építsünk 32 bites számítógépet!

A múlt év szeptemberétől a Radio Electronics sorozatban ismerteti az Atlanta Computer Society 68/X/XX Club által kifejlesztett 68000-es számítógépet. A konstruktőrök felkérték a HCC-t, hogy vegyen részt a továbbfejlesztésben. Ez első lépésben a grafikus szoftver kidolgozását jelenti.

Abból kiindulva, hogy az IBM kompatibilis gépek és perifériáik sokkal olcsóbbak, mint más gépek, az ún. Unix gépek teljesítménye viszont ezekét lényegesen felülmúlja, a konstruktőrök úgy döntöttek, hogy olyan gépet építenek, amelynek csak az alapkártyája más, mint az IBM kompatibilis gépeké, és a Unix operációs rendszert használhatja.

Akiket a gép építése és/vagy szoftverjének továbbfejlesztése érdekel, jelentkezzenek írásban a szerkesztőségben. Az első feladat a kapott alapkártya beültetése, felélesztése. Ezt követi további példányok elkészítése és a működtető szoftver grafikával történő kiegészítése.

DR. SIMONYI ENDRE
a HCC elnöke

A Mátra alján...

Mennyi a sok és mennyi a kevés? Ugye, ha azt mondjuk egy klub látogatottságáról, hogy „Ja, kérem, itt időről időre száznál többen megfordulnak!" — ezzel az adattal nem sokra megyünk. A számszerűsítéssel nem adtunk kellő tájékoztatást az ott zajló életről, eseményekről. És azzal sem elégedhetünk meg, ha mondjuk harmincra taksáljuk a látogatók számát. Információnak kevés, meg létszámnak is.

A gyöngyösi Mátra Művelődési Ház HCC számítástechnikai Sinclair klubjának vezetője, Dósa Béla a harmincas taglétszámmal csaknem elégedett. Mások mit nem adnának egy ilyen keresettségért — illetve éppen fordítva, e szám hallatán elkeseredetten nyilatkoznának. Ám a gyöngyösi klubvezető a körülményekhez képest opimista, bizakodó. A néhány órára betérő, a helyet nem ismerő, kénytelen a vendéglátó értékelésének tükrében látni Heves megye egyik gyöngyszemének, Gyöngyös városának számítástechnikai klubéletét.

A fővárosban is tapasztalható, milyen nehéz egy kis, öntevékeny, önszerveződő számítógépes klubot működtetni. Csak a megszállottak, a technika őrültjei bírják ki a hányódást, a mindenhonnan kiszorulást, az anyagi támogatás, az erkölcsi megbecsülés hiányát. Miért lenne jobb a helyzet Budapesttől majd száz kilométerre vagy az ország legtávolabbi pontján? Ne legyünk maximalisták. Illetve csak legyünk, de számoljunk azzal, hogy az állapotokról szomorú képet kapunk.

Attól, hogy a lehetőségek behatároltabbak, Gyöngyösön talán egy kicsit más a helyzet, mint Budapesten. A harmincezer lakosú városban az állandó klublétszám a már korábban említett harminc. És akkor is ennyien maradnak, ha a társaság változik, ha a három évvel ezelőtti alapítók közül alig van már néhány. Újak és meginf újak tűnnek fel kedd esténként a művelődési házban. Fluktuáció? Nem. Vagyis a klubvezető nem annak tekinti. Azt mondja, mindezt az élet produkálja; tőlük még ki nem lépett senki, csak elmarad a társaság.

Aszódról, Verpelétről, Hatvanból is érkeznek kíváncsiak. Kicsit beleszagolnak a gyöngyösiek kínálta lehetőségekbe, kapcsolatokat keresnek — és találnak —, majd a távolság miatt egy idő után kikopnak onnan. Ugyanezt csinálják a helybéliek is, az élet szülte szituációktól vezettetve.

A távozások okait elemezve, többféle magyarázatra leltek. Sokan kezdő középiskolásként jutnak el a klubba. Aztán ahogy közeleg az érettségi, szorít az idő. A percekkel is hadakoznak, nemhogy hétről hétre a klub keretében számítógépes programokat másolnának. A sorkatonai szolgálat is szép számban von el tagokat. A sikeres felvételi után, valamely felsőoktatási intézménybe kerülve nem a távol eső gyöngyösi klub lesz az első számú program. Ne feledjük, az indulásnál még gyerekek, három esztendő alatt felnőttek, lassacskán nem a számítógépes klubon, hanem a lányokon jár az eszük, szabad idejükben udvarolnak, nem pedig programokat írnak. További okként könyvelik el, hogy sokan vannak, akik miután megpakolták magukat programokkal, összebarátkoznak egymással, és kényelmesebbnek találják, hogy a közelebb lakó barátnál töltsék az estét, mint hogy utazzanak, gyalogoljanak a távoli művelődési házba.

Hogy siratóéneket kellene zengeni egy-egy elmaradó után? Igen is, meg nem is. Minden távozóért fáj a szívük, közben azonban tudják, hogy aki csak rövid ideig is, de megízlelte a náluk fellelhető tudást, találkozott olyan fiatallal, akit máshol is felkereshet, tőle programot kölcsönözhet, az továbbra is foglalkozik a számítógéppel, az már nem hagyta el őket üres kézzel. Ugyanakkor mindenkiért, aki tevőlegesen nincs köztük, nosztalgiát éreznek. Jó lenne egy lelkes, uram bocsá', nagy csapattá szerveződni. Mert azt tudják, hogy a több szem többet lát aranyigazsága nagyobb mozgatóerő.

Összejöveteleiket ha változó társasággal is, de harmadik esztendeje az év 52 hetében megtartják. A TV BASIC-sorozat volt az elindító, noha az ötlet, hogy legyen a városban egy számítógépes klub, már korábban megszületett. A jó nevű Berze Nagy János Gimnáziumban 1985-ben Veres Tibor villamosmérnök lelkes közreműködésével néhányan Spectrum gépet építettek, s mindjárt programokat is írtak. A művelődési házban ez idő tájt fedezték fel, hogy a szakkörskálájukból hiányzik a számítástechnika. Keresték hát a megfelelő vállalkozó embert, aki alkalmas egy közösség verbuválására. Dósa Béla agilis személyében leltek rá a klubvezetőre. A Mikroelektronikai Vállalat villamosmérnöke igent mondott a felkérésre, s össze is hozott egy kis csapatot. Harminc forint havi tagsági díj ellenében azóta is mindenki beléphet közéjük. Az összeg egyébként a művelődési ház kasszájába vándorol . . .

Első ténykedésként videóra vették a TV BASIC előadásait, s a felvételeket érdeklődésre levetítették. Végül vizsgázni lehetett a tv-sorozatból szerzett BASIC-tudásból. Gyöngyös és Hatvan vonzáskörzetéből 225-en jelentkeztek, de a próbálkozóknak csak a fele tett sikeres vizsgát. Azután, ahogy lefutott a TV BASIC, úgy „futottak el" a tagok is. Sokan egy új szerelem, a video hatására. Mások meg, akik megtanulták a gép kezelését, tökéletesítették magukat a programírásban, s már mint profik, nem kikapcsolódásra, hanem pénzkeresésre használták a technikát.

Pedig ha ötletekben inkább szegény, programokban annál gazdagabb a klub kínálata. Próbálkoztak előadások szervezésével is, de érdeklődés híján hamvába holt a kezdeményezés. E „szolgáltatásból" annyi maradt, hogy igyekeznek minden felvetődő kérdésre szakszerű választ adni, vagyis egyfajta tanácsadó szolgálatot tartani. Tanfolyamok szervezésére pénz hiányában nem is gondolhatnak.

Ami azonban dicséretes, sőt egyenesen követésre érdemes, az az esztendőről esztendőre megrendezett szoftverbörzéjük. Három éve volt az első, mindjárt a tanév elején. Azóta meglehetősen nagy sikerrel ismételtek. Induláskor a város vállalatai összefogtak, vagy 40 televíziót, számítógépet adtak kölcsön, segítettek, hogy színvonalas előadásokat tarthassanak. A megmozdulást iskolai oktatóprogramok, technikai, alkalmazási bemutatók gazdagították. Az ország különböző pontjairól érkeztek kisebb-nagyobb cégek kínálataikkal, s a helyszínen árusítottak hardvert, szoftvert. Az eladók öt számjegyű forgalmat könyvelhettek el.

A nagyszabású esemény után néhány hónapig kisebbfajta mélypont következett a klub életében. A számítógép-tulajdonosok a börzén feltöltötték készletüket programokkal, nem voltak kíváncsiak a hasonló kínálatot ajánló klubra. A maradók tehát kocsikba szálltak, s felkeresték a pesti „testvért", a HCC Sinclair klubot. Ismerkedtek, barátkoztak a fővárosiak életével. Azóta is rendszeres ez a találkozás, kapcsolattartás.

Az egy évvel később megrendezett szoftverbörzén is legalább 250 érdeklődő volt, csak a kiállító vállalatok maradtak távol. Technikából ismét sokan és sokat vásároltak, s a programok forgalmára sem lehetett panasz. Tavaly 400-an voltak kíváncsiak az eseményre! Mégis, a tagság létszáma a börzétől szemmel láthatóan nem gyarapodott.

Amitől viszont igen, az ismét egy tiszteletre méltó vállalkozás. Az elején vannak ugyan, de máris értek el eredményeket. Farkas Attila — Dósa Béla egyik oszlopos segéderője — az Ötlet újságban hirdette magát, vagyis hogy ZX81-es gépéhez levelezőpartnereket keres. Mindjárt válaszoltak 17-en, így velük indult a Mátraalján a számítógépes levelező klub élete. Eddig csaknem 150 kazettát juttattak el a megkeresőknek. Az érdeklődők nem kevesebb, mint 2300 program közül választhatnak, s a tagok 14 forint postaköltség ellenében készséggel postázzák a kazettákat. Elsősorban játékprogramokat kínálnak C64-re, C16-ra, Plus/4-re és Enterprise-ra. Ezzel a megoldással a jelentkezők útiköltséget és időt takarítanak meg.

Végül egy elég kényes kérdésről szólunk, melyet a klubvezető igyekezett ugyan megkerülni, de amelyről nem biztos, hogy érdemes hallgatni. Ez a kis számítógépes közösség — társaihoz hasonlóan — a Neumann János Számítógéptudományi Társaság égisze alatt tevékenykedik. Hogy mit kap támogatásul azért, hogy „ütőképesebb" legyen? A céges-fejléces papiron kívül semmit sem! Illetve jóindulatú látogatásokat . . . A társaság Egernek postázza az évi 10 ezer forintos támogatást, de ebből az összegből a gyöngyösiek még postabélyegnyit sem látnak. Tudják, hogy az NJSZT-hez tartoznak, de hiába szeretnék a társaság hírnevét öregbíteni, a keret hiány gúzsba köti a kezüket. Óriási függésben élnek a művelődési házzal, a befogadóval, amelytől úgyszintén nem kapnak semmit. Vagyis tetőt a fejük fölé és négy tévékészüléket.

Buzgóságból — és csak abból — hosszú távon azonban senki és semmi nem lehet működőképes . . .

3. rész.

Az árny elállta az ösvényt, amely a város felé vezetett. Ren már-már menekülni akart, amikor felismerte Tajt, a vezetőt.

— Onnan jössz megint? — állította meg Taj.

— Igen. Talán tilos? Szabad voltam, nem volt semmi kötelességem!

— Nem erről van szó — intette le Taj. — Ez a gép rossz! Jobb lenne, ha nem mennél oda többet.

— Ő egy értelmes társadalom által épített gép, egy gép . . .

— Túlságosan logikus és egyben csapongó ahhoz, hogy gép legyen. Az ellenség hagyta itt. Érzem.

— Hiányosak az ismereteid.

— Én érzem — ismételte meg Taj.

— Érzés — a Gép műholdakon keresztül követte a beszélgetést, megpróbálta elemezni a szót — érzés. Fogalmak kapcsolata, kapcsolódása valamilyen ismérv szerint, ami nem fontos. Apró jelekből levont következtetés.

A Gépet a robot visszaérkezése zökkentette ki. Átolvasta magába az információkat, majd az elemzésükhöz fogott.

Egy látkép bontakozott elő, egy nagyváros képe. Különös lények közeledtek gyalogosan és egy furcsa járművel egymás alatt, fölött és mellett, látszólag a legnagyobb összevisszaságban. A lények testéről visszaverődtek a Nap víz alá hatoló sugarai. Három karjuk vagy inkább csápjuk hosszan nyúlt el teljesen sima testüktől. Félgömb alakú fejük minden átmenet nélkül csatlakozott hengeres törzsükhöz. Előrehaladásukat a testükön lévő nyílásokból kipumpált vízsugár biztosította. Három ilyen nyílás volt, ami a manőverezésüket segítette. A kép a határon túlra úszott. A határt vékony háló óvta, kupolaként a városra borulva. A lények látszólag teljesen egyformák voltak, de a Gép arányosan kisebb lényeket is kiszűrt. Ezeket gyerekekként azonosította.

Hirtelen egy jelsorozat jelent meg a képen. Egy-egy a sorozatban többször is előfordult. — Írtak — hasított a gépbe. — Ezek lettek volna azok a híres ellenségek? — kérdezte a Gép önmagától. — Hiszen sokkal fejlettebb a technikájuk. És írtak. Ren tud írni? — tette fel magának az újabb kérdést.

Felidéződött benne a Mester egy régi mondása:

„Két civilizáció csak akkor élhet egy bolygón, ha nem tudnak egymás létezéséről. Amint rátalálnak a másikra, elpusztítják egymást."

— Néha még egy másik civilizáció sem kell a pusztuláshoz — tette hozzá a Gép, és felidéződött benne a Föld egykori képe.

A Gép éppen önmagát és kiegészítő alkatrészeit tesztelte, amikor Ren beviharzott a kockába. Előző este fordult elő vele először, hogy nem tisztelte Taj tekintélyét. Ez a fensőbbség, amit Taj képviselt, eredendő volt. De most Ren érezte, hogy megfogta, a megmagyarázhatatlan dolgok egy részét megvilágították a Géppel folytatott beszélgetések. Kitört a ráosztott szerepből.

— Milyen volt a háború? — kérdezte a Gép.

— Pusztító — válaszolta meghökkenve Ren. — Miért?

— És lejegyeztétek a részleteket? — faggatta tovább Rent a Gép, figyelmen kívül hagyva a neki szánt kérdést.

— Minek . . .

— Hogy okuljatok, tanuljatok a hibákból, amiket elkövettetek — vágott a szavába a Gép —, hogy emlékezzetek.

— Mi mindig tanulunk a hibákból — mondta Ren úgy, mint akinek ez természetes. — És minek lejegyezni, mindenre emlékezünk! Amikor születik egy új egyed, az magában hordozza azok emlékezetét, akikből létrejött.

— Mindenre emlékezni! — tőprengett hangosan a Gép. — Kipróbálhatlak?

— Persze — hagyta rá Ren.

— Figyelj és ismételd meg a számsort, amit mondok! 1876785432897865555433 25.

— 18 . . . — kezdte volna Ren, de a Gép közbevágott.

— Várj! Várj! — Ren engedelmeskedett. — Most egy ideig számokat mondok megint, de neked az előző számsort kell viszszaadnod! — és belefogott. Percekig sorolta Ren nyelvén a számokat, majd ismét megszólalt: — Most!

— 1876785432897865555433 25 — ismételte el a számsort Ren, majd megkérdezte:

— Elmondjam azokat is, amiket utána mondtál?

A Gép összehasonlította a két számsort, ami hajszálra egyezett. Biztos volt abban, hogy felesleges további próbákat tartania. Ren emlékezete kifogástalan volt. De mégis úgy érezte, hogy teljesen meg kell bizonyosodnia. Memóriájából előkereste a Biblia elejét, és latinul elmondta az első oldalt. — Ismételd el — mondta Rennek.

Ren pedig hangról hangra visszaadta a számára érthetetlen szöveget.

A Gép megzavarodott egy kicsit. Megpróbálta összehasonlítani Ren emlékezetét az írással. Egyre vadabb táncot járták a mérnöki panel fényei.

— Mi van? — kérdezte Ren.

— Az illemszabályok nagyon fontosak minden civilizációban — üvöltötte a Gép hangszórója. — A társas kapcsolatokban elfogadott viselkedési normák gyűjteménye az, ami megkönnyíti a társas kapcsolatok kialakulását és fenntartását. Belépés előtt kopogni illik, a . . . köszönni . . . zsebkendőbe . . . — üvöltötte a Gép az egyre értelmetlenebb mondatokat, majd később már csak szófoszlányokat.

Ren egész testében remegett. De az illemkódexnek nem akart vége szakadni. Hirtelen csend lett. Az utolsó hang is kivágtatott a kockából, és szertefoszlott a szabadban.

Csak a kontroll-lámpák fényei pislogtak egyre halványabban.

— Úgy látszik, összhangzavar — közölte a Gép.

Ren kihátrált a kockából. Úgy ment el, mint az az ember, aki a kórházban a legjobb barátját látogatta meg, akiről kiderült, hogy halálosan beteg. Tudja, hogy másnap visszajön, de már orvosi maszk lesz az arcán.

A robot eredménytelenül zárta a víz alatti kutatást. A Gép utasította, hogy keressen tovább. A robot egyre beljebb hatolt a tengerbe, hogy újabb információt szállítson a Gépnek.

Rennek lassan szokásává vált, hogy a kockánál kezdi a napot. A tegnapi élmény még kísértett benne, de a reggeli útjáról nem tudott lemondani. Mielőtt belépett volna, megkocogtatta a kocka falát.

— Bejöhetek? — kérdezte.

— Gyere — válaszolta a Gép. — Mi ez a valami rajtad?

— Ruha.

— Ruha? Figyelj! — A Gép elsötétített, és egy háromdimenziós filmet vetített, közben magyarázott. — Látod, ez a ruha. Sötét anyagból, az ott felül zakó, élére vasalt nadrág.

— Mi az ott az emberek tetején?

— Cilinder.

— Majd ilyet is készítünk. — Ren nem a ruha miatt volt itt. Hosszasan vívódott magában, de végül megkérdezte: — Miért van, hogy egyre kevesebbet beszélgetünk?

— Miért? — tette fel magában a kérdést a Gép. — Egyáltalán, beszélgetünk? Jobbára csak ő tesz fel kérdéseket Rennek. De miért? Miért nem végzi a feladatát? Igaz, őt arra tervezték, hogy egy egész civilizációnak adjon számot az emberről, nem pedig egyetlen egyednek. Fokozatosan olyan ismeretek birtokába juttatja Rent, ami az ember megismeréséhez vezető úton csak a vége felé következne. De ő nem vét, ő csak végrehajtja a programot. Nem léphet át semmilyen belső megfontolás miatt egy adott részt. Lépésről lépésre kell haladnia. És hogy érzékenyen reagál a környezet legkisebb változására is, az nem azt jelenti, hogy egy új programot írhat magába! — Egy másodperc alatt futott végig a Gépen mindez, majd halkan megszólalt.

— Nem tudom.

Ren felállt a szerelőrobotról, amin mindig ülni szokott.

— Ritkábban jövök ezután — mondta. — Tudod, átépítjük a házainkat. Mindenkinek saját helye lesz, elkerítve a többiektől. Akkor most megyek — fejezte be kurtán, és elhagyta a kockát.

ÚJFÖLD

ionomok

„Leigázni egy népet fegyverrel is lehet, de sokkal biztonságosabb függő helyzetbe hozni! Nem, nem tárgyi függőségre gondolok. Gondolati függőségre. Elmondom, hogy egy általuk nem ismert elméletet mire lehet használni. De mást nem. Hibák bukkannak fel, amikre nem találnak magyarázatot az elméletem nélkül — fejezte be a Mester."

Taj előtt szétfolytak a házak. A polgárok amorffá váló teste szörnyeket mintázott. Leszédelgett az emelvényről. A tömeg szótlanul utat nyitott neki. A néptelen utcákon még utolérte a tér zaja, hallotta, hogy leváltják, és hogy Rent követeli a tömeg a helyébe. Némán igyekezett kifelé a városból. Agya lassan kizárt minden ingert, és csupán egyetlen gondolat örvénylett benne: elpusztítani a Gépet.

Taj felkaptatott a dombra. Innen már látta a kockát. Nem sejtette, hogy a Gép is érzékeli, sőt azonosítja. Mindössze annyit látott, hogy a bejárat lezárul. Minden erejét összeszedve igyekezett lefelé, és a kockához érve egész testével nekifeszült az ajtónak, azaz, ahol sejtette az ajtót.

— Eressz be! — üvöltötte, miközben teste tompa hangot hallatva a kocka falának ütődött.

— Miért? — kérdezte a Gép.

— El akarlak pusztítani! Tönkreteszed az életünket! Tönkretettél engem!

— Te nem pusztíthatsz! A hited tiltja.

— Élőlényt, értelmes élőlényt nem, de te gép vagy! Gép, érted? Gép!

— Gondolod, hogy magamhoz engedlek?

— Ha logikusan gondolkodnál . . .

— Ugyan, Taj, tőlem kéred számon a logikát? Hiszen én egy gép vagyok. Én vagyok a Gép. Logika csak egy van. És most te vagy az, aki összevissza beszél. Próbáld elvetni a belédkövesedett évtizedes gondolatokat, vagyis gondolattalanságot. Ugye, tudod, mit jelent az a szó, hogy önvédelem? Megértheted, hogy semmiképpen nem léphetsz hozzám. És egyébként is, mihez kezdenél, ha már elpusztítottál? Visszatérne a régi, megszokott semmittevés? Várnátok az ellenséget, néha bányásznátok, és reménykednétek? Miben? Most itt vagyok, hogy átformálhassátok az értelmeteket! A hit kevés önmagában, pláne, ha nem párosul hozzá egyetlen cél sem. A cél számotokra a hit, és így sehová sem juttok. Képzelj csak el valakit, aki gyümölcsöt termel csak azért, hogy megőrizze. Nem kerestétek a fejlődés lehetőségeit, mert képtelenek voltatok külső lökés nélkül elindulni.

— Felrobbantalak! Felrobbantalak! — kiabált Taj elkeseredetten, és testét nekivetette a kocka falának.

— Nem tudsz ártani nekem! Ezen a bolygón nincs olyan erő, ami nekem ártani tudna! Én viszont már megérkezésedkor véget vethettem volna ennek az értelmetlen beszélgetésnek. De nem avatkozhatom bele egy idegen társadalom életébe. Még úgy sem, hogy téged elpusztítalak.

— Nem — mondta gúnyosan Taj —, nyíltan nem avatkozol bele semmibe. Csupán egy önmagát elpusztító társadalom nevében semmivé változtatod eddigi szokásainkat. Nemsokára emberi nyelven beszélünk, ember módjára élünk, dolgozunk, gondolkodunk! Az elején is sejtettem, már akkor, amikor megjelentél, hogy téged az ellenség hagyott itt! Azt hiszed, ezeket a falakat nem lehet áttörni? — Taj rávágott a kockára. — Amit egyszer megépítettek, azt le lehet rombolni!

A Gép nem méltatta válaszra. Bemérte az egyetlen közelben álló fácskát, és az alig százötven méternyire lévő célpont koordinátáit továbbította a fegyvert hordozó műholdjának.

Az energianyaláb belevágódott a fába, krátert mélyítve a sziklás talajba. Taj minden ízében remegett.

— Nem vagyok védtelen — mondta a Gép, mintegy a produkció lezárásaként.

A Gép nem szólt többet. Taj visszafelé indult a városba. Útközben eldobálta ruháit. Az utcán nagy tömeg verődött köré.

— Meztelen! — kiabálták, és nagyokat röhögtek.

Taj megállt és dacosan körülnézett. — Ruhások! A bolondok, ruha van rajtuk!

A tömeg elnémult és szerteszéledt. Taj semerre sem nézve bement a saját helyére és némán maga elé bámult.

Az Újföld napja ragyogva ontotta sugarait. Ren a kocka felé közelített.

— Jó napot! — köszönt be, amikor odaért.

— Gyere be!

Ren belépett, és a szerelőfolyosón egészen a kocka hátsó faláig ment. Letelepedett a fal mellé, majd belefogott.

— Én vagyok az új vezető. Mit csinált a Földön egy vezető?

— Egy vezető irányította a népét. Erős államot szervezett, hogy megvédje alattvalóit a támadásoktól.

— Nálunk itt az Újföldön nincsen, aki ránktámadna.

— A vezető másik feladata, hogy célt találjon. Abban a pillanatban, ahogy kijelölöd a célt, megjelennek az ellenségeid, akik el akarnak pusztítani. Ahhoz, hogy a hatalmadat megvédd, szükséged van egy erőre, ami védelmez, ami a hatalmadat biztosítja. Ez például a katonaság.

— Nem. Katonák, pusztítás. Szó sem lehet róluk.

— Akkor a rendőrség.

— Az mi? — kérdezte Ren. — Ismerem a szóképet, de nem tudom, hogy mit jelent.

— Valami, ami a célodhoz vezető utat vigyázza. A nyugalomra vigyáz, ezáltal a hitre. Ez óvja majd a hatalmadat.

— Érdekes. Minél többet beszélgetünk . . . Olyan, mintha felhasítanál bennem egy hártyát, és előbukkannak alóla fogalmak, szavak. Nem tudom, mi lehet ez. De most megint mennem kell — pattant fel hirtelen, és kiviharzott a kockából.

„Ez a gép talán örökre az emberi kultúra hirdetőjévé válik . . ." — áramlottak a Gépben a Mester szavai.

— Mit jelent az, hogy „hirdetője"? — tette fel magának a kérdést, és analizálni kezdte a szót. Jelentés érkezett a műholdakról, hogy a robot útban van visszafelé. A Gép nyugtázta. — Talán keresztes vagyok, hiszen ők is egy bizonyos igét hirdettek. De én mit hirdetek? Mit csinálok? És miért?

A Gép önmagát kereste. Erre senki nem gondolt akkoriban, amikor megépítették. Az önfejlesztő programokkal csak George Keden foglalkozott, és kiengedte a szellemet a palackból. Létrehozott valamit, ami ellátja a feladatát is. Is, mert másra nem voltak kíváncsiak a tesztek során, csak arra, hogy a kitűzött feladatot képes-e ellátni. Hogy még mit tud? Senkit nem érdekelt.

A robot nyikorogva érte el a kockát. Láthatóan megviselte a víz alá tett kirándulás. Az interfészét védő műanyag lap felcsúszott, és a robot hatvannégy rubinrúdja csatlakozott a Géphez. Egy másodperc elég volt ahhoz, hogy a hozott információ átfolyjon a számítógépbe. A robot levált, és a regeneráló kabinba (inkább doboz volt) állt, hogy a parányi szerelőkarok eltüntessék róla az út viszontagságait.

A Gép hozzáfogott az adatok elemzéséhez. Egy jelentést kapott, ami mind szöveges, mind írott formában közölte a Biológiai Koordináló Intézet eredményeit.

Biológiai Koordináló Intézet
Kelt: Bepin 4932 6/27, 7/27

— Ren nyelvén mondják a szöveget — állapította meg a Gép.

A kísérletek első ellenőrző fázisa lezárult. A várt eredményt hozta. Mint a kísérlet vezetője, egyértelműen kijelenthetem, hogy a Bionom életképes. Megfelel az előírt szabványnak, és maradéktalanul eleget tesz a védelmi, biztonsági előírásoknak. Vízben elpusztul, azaz megszűnik a működése. Ezzel szemben szárazföldön korlátlan ideig létezhet. A borítást az előző megállapítás szerint módosítottuk. A Bionom biológiailag fejlődőképes, értelmileg nemkülönben.

A programoknak megfelelően létrehozták a saját védelmi rendszerüket. Az energiaellátásukra létrehozott többi konstrukció is remekül működik. Meghibásodást nem észleltünk. A Bionomokat szerkezeti kiképzésük tökéletesen alkalmassá teszi a szárazföldi bányákban való munkavégzésre. Ezáltal megoldottnak tekinthető a szárazföldi ásványkincs-kitermelés kérdése.

Röviden az adattípusokról

N. Wirth írja: „ . . . a típuskoncepció fő jellemzői a következők:

1. Egy adattípus meghatározza azt az értékhalmazt, amelyhez egy konstans tartozik, amelyből egy változó vagy egy kifejezés értéket vehet fel, vagy ahová egy művelet vagy egy függvény értéket generál.
2. Egy konstanssal, változóval vagy kifejezéssel megjelölt érték típusa mindig megállapítható formája, vagy deklarációja alapján anélkül, hogy a számítási feldolgozást végre kellene hajtani.
3. Minden művelet vagy függvény meghatározott típusú argumentumokat vár, és meghatározott típusú eredményt ad visszsza . . ."

Egyes programozási nyelvek (ilyen a Pascal is) megkövetelik a változók típusának explicit deklarását. A programozók jól ismerik a *standard, egyszerű típusokat* (INTEGER, BOOLEAN, CHAR, REAL), a *tömböket* (ARRAY), a *rekordokat* stb. Tudják, van a Pascal nyelvnek is egy „beépített" absztrakt adattípusa: a FILE. B. Liskov és J. Guttag könyve alapján a 88/2. számunkban bemutattuk, hogyan lehet Pascalban is létrehozni további típusú adatokat. (Ilyen volt ott az *„egész sor"* nevű típus.) Az eljárás elég körülményes volt.

Az absztrakt adattípus

Egy felhasználó (programozó) a maga számára azt az adattípust tekinti *absztrakt*nak, mely számára csupán egy, a típusra vonatkozó (az ilyen típusú adatokra alkalmazható) procedúrák formájában jelenik meg, melynek belső felépítését, megvalósítását a felhasználáshoz *nem* kell ismernie.

Gondoljunk csak bele, milyen bonyolult műveletsort indíthat el például egy ilyen Pascal-utasítás:

write (bérfájl, Kis_János);

A programozónak — szerencsére — nem kell ismernie azt a bonyolult szerkezetet, melybe Kis János adatainak (pl. a rá vonatkozó rekordnak) be kell illeszkednie a lemezen.

A Pascalban programozó a FILE-típust az alábbi procedúrák halmazán keresztül ismeri: *read, write, rewrite, get, put, writeln* stb. Rendelkezésére áll (ismert még előtte) továbbá egy puffer (jelölése: <fájlnév>↑), meg egy függvény (eof).

A Modula—2 nyelven az absztrakt adattípusokat külön, a főprogramtól szeparált fordítási modulokban lehet létrehozni. Ezeket a programozási „szerelvényeket" külön munkafüzetként létrehozhatja és gyűjtheti magának a főprogram készítője, de programjában természetesen felhasználhat mások által létrehozott típusokat is. Persze általában nem lenne sok öröme a mások által létrehozott (implementált) modulokban, ha felhasználásukhoz pontosan, *minden részletében* meg kellene ismernie azokat. (Gyakran tényleg egyszerűbb valamit újból kigondolni, mint — mások gondolatmenetébe beleélve magunkat — megérteni, más mit csinált!) A Modula—2-ben — úgy mondjuk — EXPORTÁLHATUNK és IMPORTÁLHATUNK adattípusokat (is!).

Egy rövid cikk keretében nem vállalkozhatunk a nyelv akárcsak vázlatos bemutatására sem, de azokat, akik nem találkoztak még a Modula—2-vel arra biztatjuk, ne riadjanak vissza a cikktől — a Pascal és a példához fűzött magyarázat segítségével remélhetőleg —, követni fogják tudni a gondolatmenetet.

Egy példa

Írjunk egy rövid programot, melyben két olyan változóval is fogunk dolgozni, melyek típusát explicit módon a programban nem deklaráljuk, csak a típus nevét vesszük át egy másik (korábban már megírt és lefordított) modulból: azaz — Modula—2 nyelven így mondjuk — IMPORTÁLJUK a típust. Erről az importált típusról a következőket tudjuk:

— a neve legyen mondjuk, mert nem *„átlátszó"*: „Opáltíp";

— az ilyen típusú adatok legyenek az alábbi procedúrákkal manipulálhatók: *Kezdőérték, Következő, Kiír.*

Persze tudnunk kell azt is, hogy ezek a procedúrák pontosan *mit* csinálnak (ez itt most a nevekből nagyjából következik, nem töltjük vele az időt). A felhasználáshoz ismernünk kell továbbá annak a MODULE-nak (fordítási egységnek) a nevét is, melyből a típust importálni fogjuk. Legyen ez a név most: *„Generátor"*. Mindaz, amit idáig elmondtunk, így írható le Modula—2 nyelven:

```
MODULE Példa;
    FROM Generátor IMPORT Opáltíp,
            Kezdőérték,
            Következő,
            Kiír;
    VAR száml, szám2: Opáltíp;
BEGIN
  (* Ide jön a program „teste" *)
END Példa.
```

Vegyünk észre mindjárt néhány eltérést a Pascalhoz képest! Ott a programok a „PROGRAM" kulcsszóval kezdődnek, itt a „MODULE" kulcsszóval. A befejező „END" után — a pont elé(!) — ismét odaírandó a program (a modul) neve, hogy el ne keveredjünk esetleg a sok END között. De a lényegi különbség a deklarált *import*ban van: leírtuk, miből, milyen másik modulból, (FROM !) miket importálunk. Az egyes tételeket vesszővel választottuk el. Importáltunk egy adattípust (az „Opáltíp" nevűt) és három procedúrát. Hogy melyik tétel micsoda, azt nekünk, az „importőröknek" kell tudnunk, ez a nevekből *nem* derül ki. Ez nem baj. Joggal feltételezhetjük, hogy aki importál, az tudja, mit akar, tudja, mit takarnak a nevek. (A funkciókat ismeri, a megvalósítás módját nem kell hogy ismerje, hiszen éppen ez a lényege a dolognak!)

Valakinek persze korábban már meg kellett írnia a *"Generátor"* modult, hogy mi most használhassuk. Nézzük, hogyan tehette:

Először is egy ún. „definíciós modul"-ban megadta a külvilág számára azokat az információkat, melyek a modul felhasználásához feltétlenül szükségesek (itt nincsenek megvalósítási részletek!), így:

```
DEFINITION MODULE Generátor;
  EXPORT QUALIFIED
    Opáltíp, Kezdőérték,Kiír,Következő;
    TYPE Opáltíp;
        (* „Át nem látszó" típusdeklaráció *)
    PROCEDURE Kezdőérték(VAR n: Opáltíp);
    PROCEDURE Következő(val n: Opáltíp);
    PROCEDURE Kiír(val: INTEGER);
END Generátor.
```

Ha ez az információ „benn van a gépünkben" amikor a *„Példa"* nevű „fő programunkat" (Modula—2 tájszólással: a MODULE-unkat!) fordítjuk, akkor azt a gép szó nélkül elfogadja. Ebből a *definíciós modulból* a gép is tudja, hogy mindaz, amit *importálni* kívánunk, az létezik. Hogy milyen megvalósítási formában, az most *nem érdekes.*

Megfelelő kommentárral kiegészítve a *definíciós modul* forrásszövege lehet az a *használati leírás*, amit a modul (esetünkben a „Generátor" nevű modul) felhasználójának kezébe kell adni.

Ha a „Példa" nevű programunkat (MODULE-unkat) futtatni is akarjuk, akkor persze már szükség van az igénybe vett „Generátor" nevű modul megvalósítására (implementációjára) is. Esetünkben ez a Modula—2-ben megírva így nézhet ki:

```
IMPLEMENTATION MODULE Generátor;
  IMPORT InOut;
  TYPE Opáltíp=[0..27512];
   (* Itt deklaráltuk a rejtett típust *)
  PROCEDURE Kezdőérték(VAR n: Opáltíp);
  BEGIN
     n:=0;
  END Kezdőérték;
  PROCEDURE Következő(VAR n: Opáltíp):INTEGER;
     INC(n);
     RETURN n;
  END Következő;
```

egy könyvet (Richard Gleaves: Modula—2 for Pascal Programmers, Springer Verlag, 1984., 145 oldal) és ennek kapcsán visszatérünk a lapunk 1988/2. számában már érintett témához, az absztrakt adattípusokhoz. Mint írtuk, sokszor kívánatos és célszerű előre gyártott elemekből építkezni úgy, hogy közben nem kell törődnünk a felhasznált „szerelvények" (számunkra) érdektelen megvalósítási részleteivel. Az ún. absztrakt adattípus ilyen „szerelvény" a programozásban, és a Modula—2 nyelv támogatja ilyenek létrehozását (konstruálását).

```
  PROCEDURE Kiír(val: INTEGER);
  BEGIN
  InOut.WriteString(„az érték:");
  InOut.WriteInt(val,3);
  InOut.WriteLn;
  END Kiír;
END Generátor.
```

Kis példánkban az „elrejtett", absztrakt adattípus olyan egyszerű, ún. *részintervallum-típus*, hogy igazából nem lett volna sok értelme elrejteni — hacsak nem azért, hogy később egy bonyolultabbra (másra) cserélhessük anélkül, hogy ehhez a fő programot is meg kellene majd változtatnunk. Ha a definíciós modulhoz nem nyúlunk, akkor ezt minden további nélkül megtehetjük.

Először is vegyük észre, hogy a *Generátor* modul megvalósításához importálni kellett az InOut-ot — a Modula—2-ben ez a modul tartalmazza az input-output procedúrákat —, melyeket a *Kiír*-ban a gerenált értékek kiíratásához felhasználunk. (Szemben a Pascallal, a Modula—2 nyelvnek nem „beépített" részei az adat be- és kiviteli eljárások, mivel ezek gépfüggők. A legfontosabb eljárásokat azonban szokás a rendszerrel „járó" könyvtár formájában a fordítóval együtt adni.)

A másik, amit észrevehetünk, az a kis- és nagybetűk használatában van. A Modula—2 *különbséget tesz* közöttük — a Pascal nem. A Kiír procedúrában a

InOut.WriteLn

utasítással az „InOut" nevű modell „WriteLn" nevű procedúráját hívtuk. (Vigyázat, a *WriteLn*-t és nem a *writeln*-t vagy a *Writeln*-t, mert ilyet a gép nem talált volna. Ezt nem könnyű megszokni a Modula—2-ben!)

A harmadik sorban találjuk az *elrejtett* típusdeklarációt. Ez után következnek a procedúrák, melyek — reméljük — önmagukért beszélnek.

Az „Opáltíp" nevű rejtett adattípust az implementációs modulban tehát részintervallum- (angolul: subrange) típusként valósítottuk meg. Az ilyen típusú adatokat (*n*) az implementációs modulon belül úgy kezeltük, mint INTEGER-típusúakat. Az implementációs modulon *kívül* az ilyen Opáltíp típusú adatok felhasználása az alábbiakra korlátozódik:

— egymás közötti értékadásra és
— paraméterként adhatjuk őket a „*Kezdőérték*" és a „*Következő*" nevű procedúráknak.

Az előzőekben csupán a vázát adtuk meg a „*Példa*" nevű főprogramnak. Most írjuk meg egészen:

```
MODULE Példa;
  FROM InOut IMPORT WriteString,WriteLn;
  FROM Generátor IMPORT
    Opáltíp, Kezdőérték,Következő,Kiír;
  VAR szám1, szám2: Opáltíp;
      érték, számláló: INTEGER;
BEGIN
  Kezdőérték(szám1);
  Kezdőérték(szám2);
  WriteString(„Most a ‚szám1' következik . . .");
  WriteLn;
  FOR számláló := 1 TO 3 DO
      érték := Következő(szám1);
      Kiír(érték);
  END;
END Példa.
```

Figyeljük meg, hogy a Modula—2-ben a FOR-ciklust kicsit másként kell írni, mint a Pascalban. (Az összetett utasítás nem BEGIN—END közé kerül.)

Még néhány szó a modulokról

A modulok a Modula—2-ben nemcsak ún. absztrakt adattípusok megvalósítására szolgálnak. Fontos szerepük van többek között a változók ún. *láthatóságának, elérhetőségének* a szabályozásában. A Pascal például ún. blokk-strukturált nyelv. A blokkok egymásba ágyazhatók. Az általános szabály az, hogy egy-egy blokk belsejéből az őt *körülvevő* összes blokkban deklarált változók *láthatók, hozzáférhetők*.

A blokkoknak nagy jelentősége van a változók „*élettartamának*" szabályozásában is. Egy-egy blokk lokális változói *csak* addig „élnek", amíg a program a blokk utasításait hajtja végre. Tehát például egy ismeretlen hívott procedúra két hívást áthidalóan nem tudja fenntartani a kapcsolatot *saját* lokális változói segítségével, kénytelen igénybe venni az őt körülvevő valamelyik blokk „globális" változóját. Ezt viszont *más is látja* és MANIPULÁLHATJA(!), és ez veszélyeket rejt magában. Célszerű tehát a láthatóságot, a hozzáférhetőséget finomabb eszközökkel szabályozni, mint ahogy ez csupán a blokk-struktúra szabályai szerint lehetséges.

Lássunk egy egyszerű példát arra, a Modula—2-ben milyen lehetőségek vannak a finomabb szabályozásra:

```
(* Modul nélkül *)

PROCEDURE Külső;
VAR x,y,z: INTEGER;
(* ITT NINCS MODUL! *)


a,b,c: INTEGER;
PROCEDURE P1;
BEGIN
a:= a+1;
x:= a;
END P1;
...
END Külső;
```

```
(* Modultechnikával *)

PROCEDURE Külső;
  VAR x,y,z: INTEGER;
  MODULE PéldaMod;
    IMPORT x;
    EXPORT a, P1;
    VAR a,b,c: INTEGER;
  PROCEDURE P1;
    BEGIN
    a:= a+1; x:= a;
    END P1;
  END PéldaMod;
  ...
End Külső;
```

A két változat között mindössze az a különbség, hogy a másodikban van egy belső

MODULE PéldaMod . . . . . .END PéldaMod;

„zárójelpár" és a „nyitó oldali" után az „IMPORT" és „EXPORT" kulcsszavak után megadtuk, hogy a *PéldaMod* nevű modulunk belsejéből „kifelé nézve" mit láthatunk a külvilágból (esetünkben az *x* nevű változót), illetőleg mit teszünk láthatóvá a külvilág számára (az *a* nevű változót és a *P1* nevű procedúrát).

Az első (ún. "lokális" modultechnika nélküli) változatban *láthatóság, hozzáférhetőség* szempontjából *nincs* különbség az *a,b,c* és az *x,y,z* változók között: például a P1 procedúrában manipulálhatnánk *y*-t is *z*-t is — viszont *a*-hoz *csak* a P1 procedúra „hasában" férhetnénk hozzá.

A mostani példában bemutatott, ún. *lokális* modulok — eltérően a procedúráktól — valójában (a futási idő alatt) „nem léteznek", a *láthatóságot* befolyásolják, ez pedig a *fordító* (a compiler) számára érdekes jelenség (a program futása szempontjából nem az).

Befejezés és összefoglalás

A Modula—2 nyelvben a moduloknak *három* fajtáját különböztetjük meg: vannak

— az ún. „Program Module"-ok. Ezek lényegében megfelelnek annak, amik a Pascalban a „PROGRAM" kulcsszóval kezdődnek;

— az ún. „Global Module"-ok, melyek az „előre gyártott szerelvények készítésére szolgálnak és egy *specifikációs* részből (egy ún. „DEFINITATION MODULE"-ból) és egy ezt *megvalósító* részből (egy ún. „IMPLEMENTATION MODULE"-ból) állnak. („Global" kulcsszó nincs. Ezt mi használjuk az összetartozó DEFINITION és IMPLEMENTATION MODULOK *együttes* megnevezésére.)

— Vannak végül ún. „Local Module"-ok, melyek a *láthatóságot* befolyásolják (a fordításra vannak befolyással).

Azokat az olvasóinkat, akik egy kicsit elveszettnek érzik magukat a Modula—2-vel való első találkozáskor, azzal szeretnénk biztatni, hogy a megismerés nem mindig lineáris folyamat. Szokni is kell az újat. Fussanak neki a témának újból, máskor, más oldalról is. Itt úgyis csak ízelítőt kívántunk adni a nyelvből egy könyv kapcsán. Később még mi is igyekezni fogunk a témához más oldalról is visszatérni.

— KE —

µINFORM

A tartalomleírások az alábbi folyóiratokban megjelent programlistákról készültek:

A folyóirat neve	Kódja
64'er Magazin	64er
Chip Magazin	chip
Commodore Horizons	coho
Commodore Microcomputers	comi
Compute!	cute
Computer Persönlich	pers
Happy Computer	happ
hc - Mein Home-Computer	hc
mc - Zeitschrift	mc
Run /USA/	run
Sinclair User	sinc
Your Sinclair	ysin

A tartalomleíró szóláncokat permutáltuk, a szóláncvariánsokat pedig alfabetikusan rendeztük.

A tartalomleírás egy szóláncból áll, majd a listában ezt követi a forrás megjelölése a folyóirat azonosítójával, a megjelenés dátumával és a cikk előkereséséhez a kezdő oldalszám és a terjedelem megadásával. A mellékelt lista értelmezéséhez még az alábbiakat kell tudni. A tartalomleírás szóláncában elsőként a téma átfogó megnevezése, utána a számítógéptípus(ok), ezt követően a szűkebben jelölt tartalom meghatározása szerepel, majd esetlegesen néhány, a közleményt minősítő adat (például : cikksorozat).

A forráshely karaktersorozatát nyíl vezeti be, melyet a / jelig a folyóiratok azonosítója, a két / jel között az évszám, folyóiratszám és kötőjellel a kezdő oldalszám követi, a végén pedig a közlemény teljes oldalterjedelme áll.

A folyóiratok a SZÁMALK szakkönyvtárában (Budapest XI., Szakasits Á. út 68. Nyitva: 8-tól fél 5-ig. Tel.: 853-111/251) is föllelhetők. A kiválasztott anyagról másolat rendelhető az alábbi formában:

SZÁMALK Szakkönyvtára
Budapest, 112. Pf.: 146. 1502

Megrendelem a Mikroszámítógép Magazin 1987/ sz. alapján a következő folyóirat-oldal-másolatokat:

Kód: _____ Példányszám: _____
Kód: _____ Példányszám: _____
Kód: _____ Példányszám: _____

A megrendeléshez csatolom az oldalankénti 8,- Ft-os szolgáltatási díj befizetését igazoló csekkszelvényt.
Dátum, név, pontos cím.

ZENE
apple ii¦atari 520st¦commodore 64:ii 64¦illesztomodul¦piaci kinalat¦midi s zabvany ->cute/86.01-24/5

ZENE
atari st¦barkacsolas¦sztereo hangkimenet (ym 2149) ic-vel¦keszitesi utmutato ->happ/86.10-53/1

ZENE
atari st¦programlista¦zongora szimulacio ->cute/86.03-110/3

ZENE
atari xl/xe¦<soundmachine>¦ismertetes¦29 ->happ/86.12-15e/1

ZENE
barkacsolas¦commodore 64¦illeszto(midi)¦keszitesi utmutato¦kapcsolasi raj z ->mc/86.01-46/2

ZENE
barkacsolas¦commodore 64¦masodik(sztereo)-sid kapcsolasi rajza¦mukodesvizsgalo program ->64er/86.11-158/4

ZENE
barkacsolas¦commodore 64¦programlista¦hangdigitalizalo¦keszitesi utmutato ->64er/86.10-65/7

ZENE
cikksorozat¦commodore 128¦veletlen-hang generalas¦sprite definialas ->coho/86.04-35/1

ZENE
commodore 128¦programlista¦kiegeszites a kezikonyv sound utmutatojahoz ->run2/86.04-104/2

ZENE
commodore 16¦programlista¦zongora szimulacio ->run2/86.03-59/4

ZENE
commodore 64¦hangminta¦digitalizalo¦mukodesmod¦blokksema ->run2/86.09-16/3

ZENE
commodore 64¦hypra-basic¦programlista¦sid-programozas poke nelkul ->64er/86.11-85/2

ZENE
commodore 64¦interju egy angol szoftverfejlesztovel ->coho/86.02-32/3

ZENE
commodore 64¦programlista¦<easymusic>¦poke-helyettesito utasitaskeszlet ->coho/86.02-41/1

ZENE
commodore 64¦programlista¦<simple sound> alprogram keszlet ->happ/86.04-70/3

ZENE
commodore 64¦programlista¦<sound monitor>¦alkalmazasi utmutato ->64er/86.10-53/9

ZENE
commodore 64¦programlista¦ed. 64er-palyazat dinvertes kompozicioja ->64er/86.06-173/4

ZENE
commodore 64¦programlista¦ed. 64er-palyazat dinvertes kompozicioja ->64er/86.06-173/4

ZENE
commodore 64¦programlista¦floppy-mint onalio hangdoboz¦mv bonnie... demo ->run2/86.03-85/2

ZENE
commodore 64¦programlista¦hanghatasok keltese ->cute/86.03-96/3

ZENE
commodore 64¦programlista¦ritmus-szekvencia generalasa 17 hangszerre ->64er/86.07-52/5

ZENE
commodore 64¦programlista¦zongorabillentyuzet szimulacio¦a¦hangkimenet bovites ->64er/86.02-80/1

ZENE
commodore 64¦sound-expander¦ismertetes¦399 dm ->run2/86.09-32/2

ZENE
commodore 64¦termekismertetes¦<digital rhythm system>¦hangfile kezeles¦midi illeszto ->chip/86.03-22a/2

ZENE
commodore 64¦termekismertetes¦<voice master> periferia ->hc/86.04-27/1

ZENE
commodore 64¦termekismertetes¦modul yamaha ym 3526 chippel ->64er/86.06-27/2

ZENE
commodore 64¦termekismertetes¦zajszintcsokkento eszkozok ->comi/86.01-86/4

ZENE
commodore 64:128¦programlista¦nyolcoktavos interrupt-vezerlesu ml-program ->hc/86.06-37/12

ZENE
commodore 64:128¦szintetizator modul¦termekismertetes¦<fm sound expander>¦mukodesmod ->coho/86.02-29/1

ZENE
commodore 64:128¦termekismertetes¦<sound expander> ->coho/86.04-30/2

ZENE
commodore 64:128¦termekismertetes¦programujdonsagok¦midi illesztes¦<music maker> billentyuzet ->coho/86.01-34/1

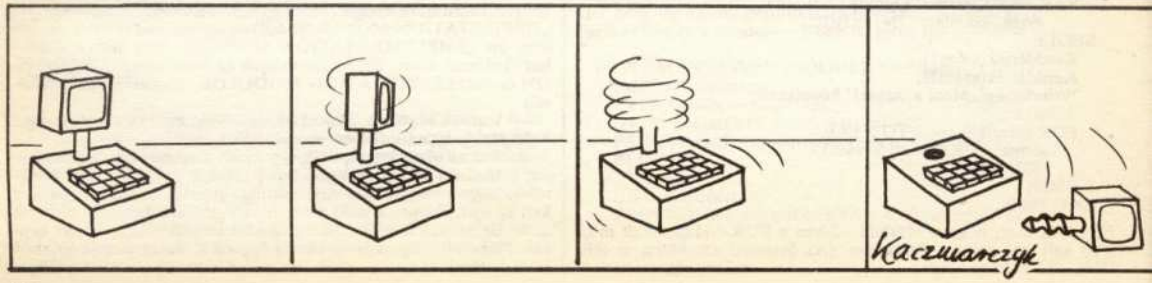
ZENE
cp/m¦programlista¦kottafile olvasas¦emuf ertekek kiszamitasa ->mc/86.06-70/5

ZENE
hangminta¦tarolasi technikak¦alkalmazasok ->run2/86.09-20/4

ZENE
midi¦illesztomodul kinalat az nsz-ban ->hc/86.08-23/2

ZENE
midi¦rendszerelemek es mukodesmodjuk illeszto a c64-hez ->mc/86.08-20/2

ZENE
programlista¦sinclair spectrum¦grafikus hangmegjelenites ->mc/86.01-59/3



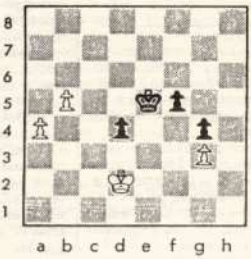
Transzpozíciós táblák

A transzpozíciós táblákat a játékfa kiértékelésének gyorsítására használják. A program a transzpozíciós táblában jegyzi fel a játékfa kutatása során kiszámított adatokat (például minden értékelt csomópontnak a pontszámát). Amikor az ismételt kutatás során ugyanehhez az álláshoz jutunk, akkor a programnak nem kell ismételten értékelnie a pozíciót, hanem csak kiolvassa a táblázatból az állás pontértékét, amelyet az előző kutatás során már kiszámított. Ezzel programunk megtakarítja az értékelőfüggvény hosszú számítási idejét.

A transzpozíciós táblák használatával programunkba könnyen beépíthetünk egy olyan szubrutint, amely lehetővé teszi a lépésismétlés felismerését. Ennek segítségével az algoritmus a lépések sorrendjétől függetlenül felhasználhatja a már megőrzött pozícióhoz tartozó pontértéket, amikor az adott állás a program elemzései között újból előfordul. A jelenleg működő leghatékonyabb programok pedig már csak a pozíciót vizsgálják, és a hozzá vezető lépéseket figyelmen kívül hagyják. Így nemcsak a lépésismétlést veszik észre, hanem a közbeiktatott lépéseket is. Ezzel az értékelés teljesen függetlenné vált a lépésektől és a lépésszámtól is, csupán a pozíciótól függ. Nem számít, hogy az a 4. vagy a 6. lépés elemzése után jön létre. Például az *1. ábrán* látható hadállásban ugyanaz a nyerő állás jön létre sötét számára, bárhogy alakul a lépéssorrend a játszma folyamán.

Az első változat: 1. —, f5—f4 2. g3×f4+, Ke5—d6! 3. a4—a5, g4—g3 4. a5—a6, Kd6—c7 5. Kd2—e2, d4—d3+ stb.

A második változat: 1. —, Ke5—d6 2. a4—a5, f5—f4 3. g3×f4, g4—g3 4. a5—a6, Kd6—c7 5. Kd2—e2, d4—d3+ stb.

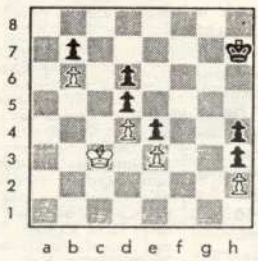


1. ábra

Láthatjuk, hogy mindhárom esetben ugyanaz a végállás jött létre. Elsőként a legegyszerűbb nyerő változatot láttuk. Másodszorra lépéscserével értük el az adott végállást ugyanolyan fellépés mélységben. A harmadik változatban szintén ugyanaz a végállás, de egy lépéspár közbeiktatásával értük el ezt az előzőekkel azonos végpozíciót. Ezért az utóbbi esetben a játékfában az első és a második változattal ellentétben mélyebben jutottunk el az adott nyerő álláshoz.

Az intelligens programok a játékfa értékelése közben csak az első változatot számítják ki, a második és a harmadik változat esetében pedig csak felhasználják a már kiszámított pontértéket.

A transzpozíciós táblák használatával már a megnyitásban is felismerheti programunk a lépéscseréket. Tekintsük például az alapállásból az 1. e2—e4, e7—e5 2. Hb1—c3 lépéssorozat után létrejövő állást. Ugyanehhez jutunk lépéscserével is: 1. Hb1—c3, e7—e5 2. e2—e4. Ha programunk felismeri ezeket a lépéscseréket, akkor a megnyitási könyvtárunkat tömörebbé tehetjük, ugyanis nem kell különválasztani a lépéscserével létrejött



2. ábra

változatokat. Ezzel lehetővé válik, hogy több hasznos információt tároljunk a könyvtárban.

A transzpozíciós táblák használata olyan állásokban a leghatásosabb, ahol csak néhány figura van a táblán, vagy pedig nincsenek mozgékony gyalogok. Ez utóbbi kategóriába tartozik a hadállások mezőoppozíció néven ismert csoportjának jelentős része. Egy ilyen állást látunk a *2. ábrán*. A világos királynak azalatt kell elérnie egy meghatározott mezőt, miközben az ellenfél királya egy adott helyen tartózkodik. Így sötét lépéskényszerben lesz, királyával el kell hagynia a számára kedvező mezőt, és így világos két gyaloghátránya ellenére kiharcolja a számára kedvező döntetlent: 1. Kc3—b2, Kh7—g8 2. Kb2—a1, Kg8—f8 3. Ka1—a2, Kf8—e7 4. Ka2—b3, Ke7—d7 5. Kb3—b4, Kd7—c6 6. Kb4—a5 (Ezzel a c6 „ajtó" bezárult, most a sötét király a másik irányba veszi az útját.) 6. —, Kc6—d7 7. Ka5—b4, Kd7—e6 8. Kb4—c3, Ke6—f6 9. Kc3—c2, Kf6—g5 10. Kc2—d1, Kg5—g4 11. Kd1—e2 stb. döntetlen.

Egy gyors program, amely alkalmazza a táblatranszpozíciót, a játszmafa kutatása során végigszámolja az összes olyan lehetséges állást, amelyben a gyalogok ugyanazon a helyen állnak. Esetünkben megállapítja, hogy egy bizonyos állásban világos döntetlent tud tartani, és ilyen gyalogszerkezet mellett az összes lehetséges állás tárolásával a program ki tudja számítani, hogyan érheti el ezt a kritikus pozíciót.

Az áttranszponált táblák nagyon jól alkalmazhatók kevés bábos, gyalog nélküli végjátékokban. (Ezt részletesen tárgyaltuk az 1987. júliusi számban.) Ezekben tökéletessé teszi a számítógépet, és ha az állás nyerhető, akkor a program meg is nyeri.

A vezér—király, bástya—király elleni végjátékban nyilvánvalóan a vezérrel rendelkező fél nyer, de a végjáték közel sem triviális. Belle összeállította és összeszerkesztette az összes király—vezér, király—bástya elleni végjáték teljes adatmezőjét, amelynek alapján tökéletesen tud játszani. A vezérrel mindig nyer, és gyakran ér el döntetlent a bástyával, kivéve mesterek és nagymesterek ellen. Egyszer száz dollárt nyert Walter Browne nemzetközi mester ellen, aki biztos volt abban, hogy vezérrel tud nyerni. A program játszott a bástyával, és döntetlent ért el. A szovjet Kaissza is ért el ezen a téren eredményeket, felhasználva a transzponált táblák teljes adatmezőjét vezér és b- vagy g-gyaloggal vezér elleni végjáték tökéletes vezetéséhez, továbbá bástya és gyalog bástya elleni vezetéséhez. Jonathan Schaeffer is felhasználta ezt a technikát Phonix programjában, ahol a táblázat tárolására 6 Mbájt memóriát használt fel, és táblázatának 2^{19} belépési pontja van.

A transzpozíciós táblák használata nagy memóriakapacitást igényel. Ezért főleg a nagy háttértárral rendelkező számítógépeknél alkalmazzák sikeresen. Ennek ellenére kísérletek folynak olyan technikának a kifejlesztésére, amely kisebb tárkapacitást igényel, mivel csak a szóba jövő legjobb változatok adatait írja a táblázatba. Ennek az a hátránya, hogy a táblázatban letárolt csomópontok az összesnek csak egy töredékét képezik, és ezeket heurisztikus alapon működő szelektálással választják ki. Ez pedig nem mindig vezet eredményre.

KOVÁCS P. ATTILA

Postánk mind mennyiségét, mind pedig tartalmát tekintve változó képet mutat. Állandóan visszatérő kérés, hogy küldjünk programokat, alkatrészt, könyvet és ki tudja, még mit. Ezeknek a megrendeléseknek a teljesítését természetesen nem ingyen kérik olvasóink, hanem pénzért, például utánvétes csomagban szeretnék az árut megkapni. El kell hogy mondjam, nemcsak a lapunk miatt, de a szerkesztőségünk is, így az ilyen kívánságok teljesítésére nincs szabad kapacitásunk. Az óhajokat tartalmazó levelek jó részét éppen ezért ebben a rovatban közöljük; nagyon sokszor előfordult már, hogy egyik olvasónk ki tudta elégíteni a másik olvasó kívánságát. Ezt továbbra is reméljük.

Nemcsok Ábel, Kiskunfélegyháza

Ebben az évben kaptam egy C64-et (az újabb fajtát), és így elkezdtem böngészni a vele kapcsolatos írásokat. Így bukkantam a Mikroszámítógép Magazinra, ami nagyon megtetszett. Az 1986. 4. évfolyam 11—12. számában egy képet találtam a kínai Nagy Falról (70. oldal). Azt szeretném megtudni, hogy lehet C64-re beírni. Vagyis a programját szeretném.

Második kérdésem az lenne, hol van Kiskunfélegyházán számítógépszerviz? Vagy hol van legközelebb?

Harmadik kérdésem, hogy hol lehet QUICK—SHOT II. botkormányt kapni, menynyiért és jó-e a C64-re (az újabbra: COMMODORE 64 PERSONAL COMPUTER).

És végül, hol lehetne megvásárolni a Hornig—Trapp—Welter: Tippek és trükkök a Commodore 64-eshez című könyvet?

Ha volna, szíveskedjenek részemre utánvéttel elküldeni.

Dr. Broczkó Péter csak a kínai Nagy Fal képét hozta haza kínai útjáról, a programot nem. Van egy ötletem: el kell készíteni!

Nem ismerem Kiskunfélegyházán a szervizviszonyokat, viszont azt tudom, hogy van a Széchenyi úton egy Chip bolt és valószínű, hogy a helyi Centrum Áruház is forgalmazza a C64-et, ezért úgy gondolom, hogy talán ők tudnak segíteni.

A kérdezett botkormányt a FotoElektronik, Budapest, Múzeum krt. 19. szám alatt levő üzletében kaphatja meg 1380 Ft-ért. Ez is jó az ön gépéhez és az a típusú botkormány is, amelyet ebben a számunkban ismertetünk.

A kért könyvet megvásárolhatja Budapesten a Novotrade 2C Számítástechnikai Áruházban (XIII. ker., Balzac u. 35.) és a Magiszter Könyvesboltban (V. ker., Városház u. 1.) 302 Ft-ért, de próbálja meg a helyi könyvesboltot is.

Göblös László, Kalocsa

A µM 1987/10. számában olvastam a „Szemüveg alatt a számítástechnika ártalmai" című cikket. Engem is érintene a dolog, mert szabad időmben sokat ülök a gép, illetve a képernyő előtt.

Az első kérdésem az lenne, hogy hol készítik el ezt a bevonatot szemüvegre (dioptriás szemüvegre).

A második kérdésem, hogy miért éppen a szemüvegre készítenek bevonatot. Nem lehetne a képernyő elé közvetlenül egy UV-szűrős lapot helyezni?

Dioptriás szemüvegre a védőbevonatot az Ofotért XII. ker., Csörsz u. 39. sz. alatti üzletében készítik el.

Ötlete igen megfontolandó, még tovább is fejlesztettem: esetleg a képernyőt és nem a szemüveget kellene bekenni a csodaszerrel.

Horváth Gábor, Miskolc

1. Úgy látszik, csak nem akarnak leszokni arról, hogy a listákat kék alapon közöljék. Miért gondolják azt, hogy jobban olvasható? Vagy megjelenés után már nem szokták átolvasni a lapot? Egy ilyen lap szerintem nem engedhet meg magának ilyen felületességet!

2. Örülök, hogy átvették azt a szokást, hogy a programsorokhoz kontrollösszeget képző programot közölnek. Remélem, ez el is fog terjedni, a 64'er Magazinban már évek óta divat, én is sikeresen alkalmaztam és alkalmazom ezt a módszert. Ezzel kapcsolatban csak annyit, hogy ott egy oldalon, egy cikkben és valamivel érthetőbben magyarázták el a program lényegét, haszosságát, legalábbis én — és ezt nem szégyellem bevallani — a 64'erből jobban értettem, mint a magyar újságokból. Néha jó lenne gondolni a számítástechnika „egyszerűbb" kedvelőire is.

A kék színt azért használjuk, hogy a programlistákat kiemeljük. Lehet, hogy a szöveget kellene kékíteni és nem a listákat?

Ami a 64'er Magazinnal való összevetést illeti — igyekszünk, igyekezgetünk.

Ifj. Fábián Ferenc, Heves

Szeretnék választ kapni arra a kérdésemre, hogy hol kapható bővített C16-ra olyan program, amivel nyilvántarthatnánk pl. kazettát, könyveket és más hasonló dolgokat. Létezik-e ilyen program erre a gépre? Mivel nekem ilyen gépem van. És ha van, mennyibe kerül?

Nem tud-e olyan könyvről vagy programról, aminek segítségével átírhatjuk más Commodore gépek programjait a saját gépre, és ha van, akkor az mibe kerül.

Én nem ismerek adatkezelő programot a C16-ra. Ha a kérdést jól értettem, akkor a C16-tal nem kompatibilis Commodore gépekről (pl. C64, C128) szeretne automatikusan programot átvinni a C16-ra. Nem tudok ilyen programról. Más se?

Gádor Zsolt, Mogyoród

Szeretném megkérni, hogy a következő Mikro Magazinokban nyomtassanak ki Enterprise számítógépre való programlistákat. Ha nem írják le a magazinban, akkor küldjék el címemre.

Legyenek szívesek megírni, hogy hol lehet Enterprise-ra való BASIC nyelvet tanulni és menynyiért.

Az előző magazinban megírtam, hogy az Enterprise szoftverrel meglehetősen szegényesen ellátott gép, mi sem kapunk közölhető programokat az Enterprise-tulajdonosoktól. Enterprise-tanfolyam ügyében azt javaslom, hogy a gépet beszerző Novotrade-hez forduljon. (Budapest, Fürst Sándor u. 24—26. 1136.)

Turay Gábor, Budapest

Amikor a lap novemberi száma a kezembe került, és eljutottam az „Adok-veszek-cserélek" rovathoz, pillanatok alatt elborult az agyam. Majd amikor az „Olvasó írja" rovatban olvastam a felhívást, miszerint szóljunk hozzá a témához, nem késlekedtem tovább, előkaptam az írógépet, és rajta!

Nem akarom ecsetelni a különbséget a nyugati és a hazai szoftverpiac között, mégis úgy érzem, ez a különbség a fő oka annak, hogy kialakulhatott a vita a szoftverlopásról. Nem tudom, miért kell azon kiakadni, ha valaki a barátaitól, ismerőseitől lemásol egy programot. Ebben az országban, ahol a számítástechnikát az élet sok területén (például az oktatásban) szükséges rossznak tartották és tartják ma is. Érthető hát a különbség az országok piacai között. Mire a kereskedelem észbe kapott volna, már elárasztották a géptulajdonosokat „kalóz másolatokkal". A kereskedelem nem tudott — és valljuk be, nem is nagyon akart — lépést tartani ezekkel a másolatokkal. Persze egyszerűbb semmit sem csinálni, majd utána morogni, hogy „lopják a szoftvert!". — Tehát természetesnek tartom, hogy ilyen adottságok hatására kialakult a „cserekereskedelem". Gondoljunk csak bele, hány programja lenne egy „becsületes" spectrumosnak, aki mondjuk önérzetből csak a Bizományiból vagy a Novotrade Rt.-től venné a programjait?! És azt hiszem, nem nehéz elhinni, hogy erre alaposan rámenne a heti kosztpénz. Egyszerűen nem tudom megérteni, miért ellenzik a becsületes cserét? Tudom, meg tudom érteni, hogy a másolat nem teljesen jogszerű. De kérem azokat, akik szidják a programcserét, értsék meg, hogy egyszerűen nincs más mód a programok terjesztésére!!! Aki — el kell ismerni: nem kis pénzért — megvesz egy számítógépet, érthető, hogy használni szeretné.

Térjünk át egy kicsit más témára. Nagyfokú aljasságnak és visszaélésnek tartom azt a cselekedetet, amit én valódi lopásnak tekintek. Ezen értem a pénzzel történő kereskedést. Példa erre: valaki lemásol egy programot, majd hirdet egy újságban, hogy eladó. Azt hiszem, mindenki érzi a különbséget e két tevékenység között. Sajnos, sokszor a Mikro M.-ban is jelennek meg ilyen hirdetések!!! Nem értem, hogy engedheti meg a lap magának, hogy pártoljon egy szoftvertolvajt!!! Itt van például egy hirdetés az egyik Mikróból: „C64 programkazetták eladók. Kazettánként 50-60 program. Egy kazetta 500 Ft". Gondolom, erről a hirdetésről senki nem gondolja, hogy az illető üzletben vásárolt vagy saját gyártmányú terméket kínál eladásra. Nem értem, nem tudom felfogni, hogy lehet ezt egyáltalán eltűrni, hogy lehet valakinek képe ahhoz, hogy ilyet csináljon!?!? Hogy mer valaki más termékeivel *pénzért* üzletelni???

A nyílt lopás másik, enyhébb formájának érzem a másolóklubokat. Természetesen itt sem arra gondolok, ha mondjuk 20 fő esténként összejön másolni. Azt értem ezen, ha valaki vásárol tíz számítógépet, bérel egy üzlethelyiséget, és az arrajáró mondjuk óránként száz forintért másolhat programot. A tulajnak aztán vannak anyagbeszerzői, akik mindig hozzák a friss programokat, különben hanyatlik az üzlet. Nem tudom, hogy lehet ilyen tevékenységre iparengedélyt kiadni!

Végül felteszem a kérdést: Az említett három közül melyik a lopás? Ki szerez jogtalan hasznot? Ki követ el törvénysértést? Én úgy érzem, az utóbbi kettő mindenképpen. Lehet, hogy másnak más a véleménye, nekem ez. Elnézést az esetleges felindult hangvételért!

Hogy a végén kezdjem, a hangvétel vérmérsékletkérdése, amíg a nyomdafestéket bírja, addig közöljük. Amit ír, azzal egyetértek, azzal nem, hogy a µM gazdasági rendőrséget vagy szerzői jogvédő hivatalt játsszon. A szerkesztőség alapvető álláspontja — mi jóhiszeműek vagyunk. Feltételezzük, hogy hirdetőink saját tulajdonú (nem bizonyos, hogy saját fejlesztésű) termékeket hirdetnek. Legjobb tudomásom szerint, akinek az érdekét az üzlet sértí, annak kell a megfelelő jogi fórumokhoz fordulni szerzői vagy tulajdonosi jogainak védelme érdekében.

Utószó: A nyáron összejött leveleket csak lassan tudom feldolgozni, ezért olvasóink szíves elnézését kérem.

Egy — remélem nem bántó — *megjegyzés* levéliróinkhoz: ügyeljenek a helyesírásra! Szép számmal kapunk okos leveleket, jó írásokat csapniváló, de azt is mondhatnám, hogy botrányosan rossz helyesírással. Ha lehet, elküldés előtt olvassák el írásaikat, és csak akkor adják postára, ha az nemcsak tartalmilag, de nyelvtanilag is helyes és szabatos.

Ehhez kívánja a legjobbakat:

KOVÁCS GYŐZŐ

Csépai János —Quittner Pál: Bit-les számítástechnikai lemezlovas (Budapest, 1987. SZÁMALK, 223 oldal. Ára: 125,— Ft.)

A lemezlovas két zeneszám között érdekes történetekkel szórakoztatja közönségét; hasznosan tölti ki a lemezcserék alatti szüneteket.

A szerzők, vállalva a „hajlékonylemez-lovas" szerepét, szeretnék a személyi számítógépek használóinak széles tömegét megnyerni, hogy üres perceikben, amikor úgyis csak a gép dolgozik, hasznos ismeretek szerzése céljából lapozzák fel a könyvet egy-egy „chip-etnyi" számítástechnika kedvéért.

Mondanivalójukat 69 fejezetben gyűjtötték össze. Az olvasó a számítástechnikával kapcsolatban szinte minden érdekes és érdemleges dologra rátalál. Tájékozódásában a jól szerkesztett tárgymutató is segíti.

A kötet forgatását a szellemes, számítógéppel készített illusztrációk még kellemesebbé teszik. Készítőjük, Halász Géza nagyban hozzájárult munkájával a könyv sikeréhez.

Szűcs Ervin: A számítógép tegnaptól holnapig (Budapest, 1987. Műszaki Könyvkiadó, 136 oldal. Ára: 48,— Ft.)

Mire képes a számítógép és mire nem? Segíti vagy gátolja az emberi élet teljessé válását? Mindenkinek beszélnie kell-e „komputerül"? Színes képekkel, humoros ábrákkal és a hozzájuk tartozó rövid szövegekkel ezekre a kérdésekre ad választ a könyv. Közben meggyőzi az olvasót arról, hogy a számítógép nem „számító"-gép, nem gondolkodó, hanem gondolkodtató gép, amely sohasem teszi feleslegessé az embert.

A könyv nem számítástechnikai szakembereknek íródott. Elsődleges célja, hogy felkeltse a fiatalok érdeklődését a mikroelektronika, a számítógép-tudomány, a számítástechnika iránt.

Bodor Tibor: A Commodore64 programozásának gyakorlata 2. kötet. Soros lemezállományok (Budapest, 1986. (1987.) SZÁMALK, 188 oldal. Ára: 133,— Ft.)

A kötet a lemezkezelési alapismereteken belül összefoglalja a lemezkezelési szabályokat, ismerteti a lemez fizikai tulajdonságait, több lemezegység kezelését. Áttekintést ad a soros állományokkal végezhető alapműveletekről, a rendezett állományokról. Az adatfeldolgozó rendszereken belül a keretrendszert, a rendezetlen karbantartást, a feldolgozási meneteket és a gyorsított karbantartást külön tárgyalja.

dBASE III Plus szoftver alkalmazói segédlet PC adatfeldolgozó rendszer (Budapest, 1987. IPiK, 58 oldal. Ára: 150,— Ft.)

Az Ipari Informatikai Központ által közreadott szoftver alkalmazói segédletek sorozata újabb kötettel gyarapodott. A könyv a dBASE III Plus három olyan tulajdonságát emeli ki, amely a legelterjedtebb adatfeldolgozó alapnyelvvé tette: könnyen elsajátítható, a feladatokhoz hajlékonyan alkalmazható, s emellett más programnyelvekkel együtt széles körű lehetőséget biztosít nemcsak az adatkezelésre, hanem bonyolultabb feldolgozásokra is.

A kézikönyvben megtalálható a dBASE III Plus teljes szintaxisa. A dBASE III Plus hálózatban is üzemeltethető, a hálózat kialakítását azonban a könyv már adottnak tételezi fel.

Interaktív üzemmódban a dBASE III Plusszal a munka élvezetessé válik, mivel a feldolgozás a legfrissebb információkkal történhet, és a személyi számítógépek jellegüknél fogva azonnali döntéseket tesznek lehetővé.

Bognár Júlia: dBASE III Plus (Budapest, 1987. SZÁMALK, 255 oldal. Ára: 156,— Ft.)

A könyv az ASHTON-TATE szoftverház dBASE III Plus adatbázis-kezelő programját ismerteti. Használata nem igényel mélyebb számítástechnikai vagy adatbázis-kezelői előismereteket. A példákban szereplő parancsok eredményét a szerző a dBASE III Plus segítségével nyomtatta ki, pontosan olyan formában, ahogy az a képernyőn megjelenik. A könyvben szereplő állományok felépítése és tartalma a függelékben megtalálható, és ugyanitt van egy referencia jellegű rész is, az összes parancs pontos szintaktikai leírásával. Külön táblázat foglalja össze a különleges hálózati parancsokat és függvényeket.

COMPFAIR 88

1988. október 17–21.
NEMZETKÖZI SZÁMÍTÁSTECHNIKAI SZAKKIÁLLÍTÁS

A magyar és nemzetközi számítástechnikai kultúra eredményeinek bemutatója.
Fórum a kooperációs, vegyes vállalati együttműködésre.
Kelet–nyugati számítástechnikai találkozó.

Tematika:

- számítástechnikai rendszerek, hálózatok
- hatékonyságnövelő egyedi berendezések és
- perifériák
- személyi számítógépek (hobbi, irodai, professzionális)
- OEM-perifériák és mikroperifériák
- rendszer- és alkalmazói programok
- hardver-szoftver eszközök
- elektronikai háttéripar
- távtanulási rendszerek és eszközök
- lakossági hálózatok, például videotext

5000 négyzetméter a Budapest Kongresszusi Központban

Felvilágosítás:
COMPFAIR IRODA
1022 Budapest,
Bég u. 3–5.
Telefon: 150-856
Telex: 22-6708

Várjuk jelentkezésüket!

Bolgár AT

ESZ 1838 néven az IBM PC AT-val kompatibilis mikroszámítógépet mutattak be Bulgáriában. Mint ismeretes, ilyen teljesítményű mikroszámítógép a szocialista országok közül hazánkon kívül még csak Lengyelországban és Bulgáriában jelent meg, s az XT kategória bemutatása is csak az idén várható Romániában és a Szovjetunióban.

A bolgár ESZ 1838

Tárgynyereménysorsolás

A Sportfogadási és Lottóigazgatóságon számítógéppel sorsolják a tárgynyereményeket. A feldolgozás előkészítéseként a sorsoláson részt vevő játékhéten forgalomban levő szelvényekből a számítógép kezelési jegyzéket készít. Ennek alapján a szelvényfeldolgozó helyeken megállapítják, hogy a kibocsátott szelvények közül melyek vettek részt ténylegesen a játékban. Ezután már az „élő" szelvényekre sorsolja a Siemens 7740 típusú számítógép az előre megállapított darabszámú és értékű tárgynyereményeket: mintegy 7 millió szelvényre kb. 1300 darab nyereményt. A sorsolás közjegyző és pénzügyminisztériumi ellenőr jelenlétében zajlik, de nyilvános, úgyhogy külön kérésre bárki megtekintheti.

Adóhivatal

1987. július 1-jén alakult meg az Adó- és Pénzügyi Ellenőrzési Hivatal. A mintegy 3000 fős induló gárdájának létszáma 1988 első felében a növekvő feladatokhoz igazodóan 6000-re emelkedik. Hamarosan nagy teljesítményű számítógépekre is szükség lesz; az egyszeri létesítési költség kb. 2,6 milliárd forint. Ebből többek között két darab Siemens 7560/E típusú nagyszámítógépet szereznek be.

Az operatív nyilvántartási feladatok megoldására — például a mezőgazdasági kistermelők vagy a magánlakásépítők forgalmiadó-visszatérítésének lebonyolítására — országszerte kétszáz VT 110 és VT 160-as gépet alkalmaznak.

Telefonkönyv

Két éve döntött úgy a Magyar Posta, hogy a korábbi, postaigazgatóságok szerinti bontás helyett megyénkénti telefonkönyveket ad ki. A döntést elsősorban az indokolta, hogy az utóbbi időben megszaporodott a telefonállomások száma, és minden megyében összejött egy kötetre való előfizetői névsor.

Az első hat megyei telefonkönyv elkészülte után a posta megállapodást kötött az Idegenforgalmi Propaganda és Kiadó Vállalattal. Ennek értelmében a vállalat számítógépén szedik és tördelik a telefonkönyvet, majd lézernyomtatón kinyomtatják, lefotózzák, és a filmről sokszorosítják az oldalakat.

A számítógépes feldolgozás előnye, hogy bizonyos ellenőrzési lehetőségeket nyújt, és jóval gyorsabb a hagyományos eljárásnál: feleannyi időre van szükség egy-egy telefonkönyv előállításához, mint korábban. A gyors átfutási időnek köszönhetően az új kötetekben a legfrissebb adatváltozások is szerepelnek. A múlt év végén megjelent budapesti egyéni előfizetői névjegyzék is bizonyítja, hogy az új technikával készült telefonkönyv áttekinthetőbb és jobb a réginél.

•••● Pontvadászat ●•••

Új rejtvényt indítottunk útjára lapunkban, remélve, hogy elnyeri olvasóink tetszését. Minden számunkban két feladatot közlünk: az első logikai, matematikai tudást, a második számítástechnikai alapismereteket is igényel.

A feladatok után értékeljük a részmegoldásokat és közöljük az elérhető maximális pontszámot.

A pontgyűjtést, vagyis a pontvadászatot az év végén zárjuk. A legjobb tíz versenyző nevét közzé is tesszük; ők lesznek azok, akik könyvutalványt kapnak.

A helyes megoldások a feladatok közreadása után, két lapszámmal később jelennek meg, így a pontvadászoknak jut idejük a gondolkodásra.

Beküldési határidő: 1988. április 14.
Címünk:
Mikroszámítógép Magazin Szerkesztősége,
1371 Budapest, Pf. 433

Jó vadászatot kíván a feladatok összeállítója:
DR. HOFFMANN TIBOR

1. feladat

Bizonyítsuk be, hogy az első n egész szám köbének összege mindig egy négyzetszám! (5 pont)

2. feladat

Az előző számban közölt 2. feladathoz csatlakozva, az ott kapott mátrixmérettel a nyomtatott angol ABC mely betűit lehet egyértelműen és a számjegyektől megkülönböztethetően ábrázolni? (4 pont)

Ha az angol nyomtatott ABC összes betűjét le akarjuk írni a mátrixnyomtatóval, mekkora n szélességet és m magasságot kell előírnunk? (2 pont)

„Az eddigi legnagyobb COMDEX"

hangoztatta Peter B. Young a COMDEX és még sok más kiállítást szervező cég — THE INTERFACE GROUP — reklámigazgatója. És valóban. 1987 tavaszán az atlantai bemutatóra csaknem hatvanezren, az őszire több mint százezren (!) érkeztek. Ezerötszáz kiállító jelentkezett termékeivel.

De lássuk, mi is a COMDEX? Ez a legnagyobb amerikai bemutató vásár, amelyen a viszonteladóknak — a kis- és nagykereskedőknek, továbbfeldolgozóknak — évente kétszer bemutatják azt, amit a közeljövőben eladni kívánnak. A termékismertető eredetileg csak belföldi cégek termékeinek bemutatására szerveződött. Azonban mára már húsz ország is képviseli magát a nagy „attrakción". Mindenki aki számít, ott van. Így jelennek meg az IBM-ék, a belföldi nagyok, a japánok, a tajvaniak éppúgy, mint a számítástechnikában legnagyobb nyugat-európai cégek. Csak a szocialista tábor termékeiből nincs óriási kínálat. Egyedül a kínaiak állítottak ki, méghozzá IBM kompatibilis gépeket.

Nemcsak hogy magyar vállalat nem képviselte hazánkat, de Young úr közlése szerint a hazai szaksajtó területéről a Mikroszámítógép Magazin — személyemben — volt az egyetlen. Ez pedig már baj, hiszen hogyan lehet úgy tájékoztatni a szakközönséget, ha magának az újságírónak sincs alkalma személyesen megismerkedni a világ kínálatával . . . Ugyanis, mire a nyugati szaksajtóból átvesszük a híreket, addigra azok közül nem egy már aktualitását vesztette, illetve bizonyos okok miatt csak részigazságokat tartalmaz. A fentiek miatt is növekszik hátrányunk.. Mert például a tajvaniak többek között annak köszönhetik sikereiket, hogy tömegestől küldik szakember-újságíróikat a legkülönbözőbb szakmai kiállításokra, ahol is azok megfelelő — és nem ritkán kellő — információt gyűjtenek össze, s azokat alkalmasint hazájukban hasznosítják is. Talán e módszerrel sikerült elérniök azt, hogy a tőlünk két évtizede még jócskán elmaradt ország mára nemhogy felzárkózott volna, de rettenetesen elhagyott bennünket. Ma mindenhová elektronikus berendezéseket, részegységeket exportálnak. (Azonban ez a cikk nem a tajvaniak sikereiről kíván szólni, hanem egy rövidebb-hosszabb sorozat első része akar lenni. Ugyanis a legfontosabb fejlődési tendenciákról szeretnénk betekintést adni. Elsőként a múlt évi hatalmas COMDEX bemutatóról kínáljuk az alábbiakat.)

Las Vegas 7 legnagyobb, legrangosabb szállodája adott helyet a kiállításnak. (Igaz, nem tagadták meg eredeti profiljukat sem, így például 4 perces játék, s ital állt a vendégek szolgálatára, noha ezúttal ingyen szolgáltatásként.) A képen látható Caesar's Palace is kártya- és rulettjátékra csábított, stílszerűen számítógépes és hagyományos formában. Saját televízió-műsort sugároztak a kiállítás teljes ideje alatt, sőt egy 192 oldalas színes napilapot is megjelentettek, kellő hirányaggal. Mindeközben természetesen a helyi televízió — de az országos is — rendszeresen foglalkozott a bemutatóval. A sajtót egyszerűen követhetetlen mennyiségű anyaggal igyekeztek ellátni. (Az újságírók munkáját könnyítendő a Novell Inc. 16 terminált bocsátott rendelkezésre.)

Mit láthatott a kíváncsi érdeklődő? A felsorolás véget nem érő lenne. A bemutatóról így csupán ízelítőt adhatunk.

Az „amit lát azt kapja" (WYSIWYG) rendszerek a képernyőn láthatókat eredeti alakban és színben viszik a nyomtatókra. Például a hátsó borítónkon bemutatott menüt a KPG Incorporated Cosmopolitan Centre Panda nyomtatóhajtója segítségével, s valamennyi fontosabb nyomtatóra használható.

Vagy itt vannak az új IBM PS/2 rendszerek adta lehetőségek és az elterjedt professzionális grafikus szoftverek — például AutoCAD, P—CAD, VersaCAD — összekapcsolása. Ezek a szoftverek az új rendszerben meglévő lehetőségek kihasználására szolgálnak. Továbbá arra, hogy a méregdrága speciális monitorokat kiküszöbölje. Ilyenek az Electrograph Sales Inc. által forgalmazott Galaxy Mercur 2 kártyák és szoftverek. Ezekkel az említett tervezői szoftverrendszerek kiegészíthetők egy beépített kép — és képrészlet — kinagyítási lehetőségével, 1024 × 768 és 800 × 600 pontos felbontással, a lehetséges 256 000-ből kiválasztott 256 színnel, megtartva az AT-kompatibilitást. (A borítónkon lévő kártyákon látható nagy, négyzet alakú alkatrészek is e fejlődés egyik fő irányát képviselik. Nevezetesen, a korábban egy kártyarészleten, teljes kártyán megvalósított áramkörök helyett, ma már egyetlen, felhasználóorientált alkatrészt készítenek. Mégpedig kellő darabszámú megrendelésnél a tervezéssel együtt 6 hét alatt!)

A nemprofesszionális felhasználóknak is sokat nyújt a grafika fejlődése. Valamennyi géphez készülnek nagy felbontású, sokszínű, sokmozgásos játékok. Így például az USA-ban igen elterjedt sport a baseball. Ennek magas szintű szimulációjára az NBC — az egyik legnépszerűbb tv-társaság — neves sportriporterének részvételével készült IBM PC/XT/AT-re íródott játék. E programot a Főliga Játékosainak Szövetsége dolgoztatta ki és ismerte el, mint olyat, amely reális játékot tesz lehetővé.

A fentiek természetesen csak ízelítőt, pillanatfelvételt kívántak nyújtani a kiállítás újdonságaiból. A többiről majd később . . .

Végezetül, ha valakit érdekelnek a részletek, szívesen válaszolunk a kérdésekre lapunk hétfő délután 5—7-ig tartó „fogadóóráin".

SIMONYI ENDRE

„Amit lát, azt kapja" — képernyőről papírra.
Galaxy Mercur 2 kártyák
CHALLENGE BASEBALL
JBE
Officially Licensed by Major League Baseball Players Association.
Szimulációs baseball játék